STATISTICAL AND GEOMETRIC METHODS FOR SHAPE-DRIVEN SEGMENTATION AND TRACKING

A Thesis
Presented to
The Academic Faculty

by

Samuel Dambreville

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
April 2008

STATISTICAL AND GEOMETRIC METHODS FOR SHAPE-DRIVEN SEGMENTATION AND TRACKING

Approved by:

Professor Patricio Vela,
Committee Chair
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Marc Niethammer
Department of Computer Science
University of North Carolina Chapel Hill

Professor Allen Tannenbaum, Adviser
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Yucel Altunbasak
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Anthony Yezzi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 29 Februrary 2008

DEDICATION

To my father,

Docteur Jean-Marie Dambreville (1953-2004),

in loving memory.

ACKNOWLEDGEMENTS

During my graduate years (notably), I was blessed with many relationships with extraordinary people. I would like to take this opportunity to thank some of them whose beneficial influence led me to undertake and pursue my PhD degree. In particular, I am forever indebted to

- Prof. Allen Tannenbaum (a.k.a. Jedi Master Tannenbaum) for his kind supervision, his inspiring leadership and for enabling me to improve myself in more areas than I could possibly enumerate.
- Prof. Anthony Yezzi (a.k.a. Jedi Master Yezzi) for his patience, his sharp guidance and his most valuable influence on my research.
- Prof. Marc Niethammer for teaching me the tools to start my research on solid bases and for his most appreciated friendship.
- Prof. Patricio Vela for his interest in my work and his cheerful encouragements.
- Prof. Oskar Skrinjar for transmitting the passion of image processing and computer vision to me.
- Prof. Yucel Altunbasak for accepting to be a member of my PhD committee.
- Dr. Yogesh Rathi for his energy, for being such an inspiring research teammate and for his precious friendship.
- Romeil Sandhu for his enthusiasm, his sense of humor as well as for being such a valuable research partner and friend.

- the other members of the Biomedical Imaging Lab: Ponnappan Arumuganainar, Yi Gao, Eli Hershkovits, Shawn Lankton, Xavier LeFaucheur, Jimi Malcolm, John Melonakos, Vandana Mohan, Delphine Nain, Eric Pichon, Gallagher Pryor, Tauseef Rehman, Julien Traisnel, Amanda Wake, Yan Yang and Lei Zhu, for our multiple and fertile exchanges as well as for their contribution to the joyful yet studious atmosphere in the lab.

- the members of the "College of Management @ Tech family" for the inestimable learning experience they provided and the vivacious energy they communicated to me. In particular, I am thankful to Ann Scott, Prof. Goutam Challagala, Peter Vantine and Robert Burgess.

- my CM2 teacher, Christian Boyer, for making me understand the importance of studying.

- my friends from Atlanta and elsewhere for enlightening my days. In particular, I am thankful to Laurence Aubignat, Arnaud Bistoquet, Julien Caritey, Francoise Chen-Chi-Song, Karine Fontaine, Luc Gacougnolle, Nathalie Gomez, Corinne Huet, Francois Kulling, Samson Lim, Gerome Peter, William Sommer and Gelsy Torres-Oviedo.

- my uncle, Philippe Belon, for his help during difficult times.

- the other members of my family, for their support.

- Kelly Henry, for her love.

- my mother, Colette Benard, for everything.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

Computer Vision aims at developing techniques to extract and exploit information from images or other multidimensional data. The successful applications of computer vision approaches are multiple and have benefited diverse fields such as manufacturing, medicine or defense.

Some of the most challenging tasks performed by computer vision systems are arguably segmentation and tracking. Segmentation can be defined as the partitioning of an image into homogeneous or meaningful regions. Tracking also aims at extracting meaning or information from images, however, it is a dynamic task that operates on temporal (video) sequences. Active contours are ubiquitous tools in computer vision applications and have been proven to be quite valuable at performing the two aforementioned tasks. The active contours framework is an example of variational approaches, in which a problem is compactly (and elegantly) described and solved in terms of energy functionals.

The objective of the proposed research is to develop statistical and shape-based tools inspired from or completing the geometric active contours methodology. These tools are designed more particularly to perform segmentation and tracking. The approaches developed in this thesis make an extensive use of partial differential equations and differential geometry, notably, to address the problems at hand. Most of the proposed approaches are cast into a variational framework. The contributions of this thesis can be summarized as follows:

1. An algorithm is presented that allows one to robustly track the position as well as the shape of a deformable object. The technique combines state of the art

statistical and shape-driven approaches in a novel and computationally efficient manner.

2. A technique for the segmentation of regions is proposed. A cost functional is defined that extracts image information, adopting a shape-driven point of view. The resulting approach uses both geometric and statistical information in images to separate regions.

3. Diverse frameworks are introduced for including prior knowledge on shapes in the evolution process of geometric active contours. Different kernel-based methodologies are proposed, tested and compared in the context of shape learning.

4. A novel framework is proposed that combines statistical information extracted from images with shape information learned a priori from examples. The approach fuses image information and shape knowledge in a meaningful and homogenous manner leading to robust segmentation performances.

5. A technique is proposed to jointly segment a 3D object of arbitrary shape in a 2D image and estimate its 3D pose with respect to a referential attached to a unique calibrated camera. The proposed method exploits the prior knowledge of the 3D shape of the object of interest.

6. A novel methodology for the non-deterministic evolution of curves is proposed, based on the theory of interacting particles systems. The proposed framework can be used to provide a stochastic interpretation of geodesic active contours, of interest for segmentation tasks in computer vision.

CHAPTER I

INTRODUCTION

1.1 Computer Vision, Segmentation and Tracking

Computer vision aims at developing techniques to extract, learn or exploit information from images, sequences of images or other multidimensional data. The ultimate goal of computer vision is to build machines or systems that "see". The verb "see" is understood here not only in its passive sense, as the process of acquiring images of the world, but also in its active sense, as the process of interpreting or establishing meaning from these images to guide decision or action. Hence, computer vision can be seen as a subfield of artificial intelligence concerned with cognition processes that arise from visual perception. Computer vision is also related to the field of biological vision. In biological vision, real vision systems of humans and various animals are studied, to understand and build models of how these systems are implemented in living organisms. Computer vision, on the other hand, studies and describes technical vision systems, which are implemented on computers or machines (either in software or hardware). There is more and more interdisciplinary work between biological and computer vision that benefits both fields. However, most of computer vision frameworks are developed adopting a purely mathematical point of view, instead of a biological one. Indeed, most standard methods in the field of computer vision have been developed from rigorous mathematical approaches ranging from statistics, optimization or geometry, see for instance the textbooks [42, 56, 90, 8, 45, 81, 87]. Although building an artificial system as versatile and efficient as the human eye-cortex system is still an active domain of research, many specialized systems have been developed that effectively perform specific tasks under controlled conditions.

The successful applications of computer vision are multiple and include the monitoring or control of manufacturing processes (e.g., quality control, guidance of robots...), the detection and surveillance of events (e.g., people counting, traffic control...), or the interaction with machines (e.g., computer interaction, face recognition...). The two fields in which computer vision applications are the most advanced and successful are probably medicine and defense:

- In medical applications, the imaging modalities are diverse and encompass X-ray, ultrasonic, CT and MRI scans. These diverse modalities yield very different types of "images" (or temporal sequences) that can be either 2-dimensional (2D) or 3-dimensional (3D) and live either in the gray-scale, color or even tensor spaces. A few examples of tasks, which can be accomplished from such image data, using computer vision techniques are the detection of tumors, arteriosclerosis or other malign changes. computer vision algorithms can also be useful to provide measurements of tissues dimensions, organ strains or blood flow, notably. Hence, computer vision can provide valuable tools to assist medical doctors and surgeons in performing diagnosis or surgery [6, 88]. In addition, novel developments also support medical research by providing new information about anatomical structures (e.g., the structure and connectivity of the brain [84, 76]) or about the quality and effects of medical treatments, for example.

- In military applications, the imaging modalities are also extremely diverse and encompass satellite, laser or diverse types of cameras. These diverse modalities also yield very different types of images that can live either in the gray-scale, color or infrared spaces, as well as be depth maps or range images. The obvious examples where computer vision techniques can be used are the detection of enemy soldiers, vehicles or installations and the guidance of missiles to a designated target. Also, modern military concepts, such as "battlefield awareness,"

can benefit from the use of computer vision algorithms to combine and process information from multiple imaging sensors to reduce complexity, increase reliability and help strategic decisions making.

Some of the most challenging tasks performed by computer vision systems are arguably segmentation and tracking. These two techniques are oftentimes the crucial tasks underlying the diverse applications alluded to above.

Segmentation can be defined as the partitioning of an image into homogeneous regions [118]. The homogeneity of each region is measured in terms of pre-determined criterions or computed characteristics. The typical result of segmentation is a set of regions, which taken together cover the entire image domain. Pixels belonging to a certain region are similar to each other (whereas they are dissimilar to pixels belonging to other adjacent regions) with respect to a certain criterion. The goal of segmentation is to provide a simplification of images and to extract meaning or information. The typical application of segmentation is to separate an object of interest from the background in a given image.

Tracking is a very similar task to segmentation in the sense that its goal is to extract meaning or information from images. The main difference, however, is that tracking is a dynamic task that usually operates on (temporal) sequences of images, whereas segmentation usually operates on a given static image. The typical application of tracking is to follow the evolution of an object in a video sequence. An efficient and reliable segmentation technique can be used on successive images of a sequence to perform tracking. Alternative approaches combine prior information from a (temporal) model and information obtained online from successive images. These tracking approaches are usually based on filtering techniques such as Kalman filter [62] or particle filter [34].

1.2 Variational Approaches, Active Contours and Level Sets

Variational methods, see [81] and the references therein, received a lot of attention is the past few decades in the computer vision and image processing communities. Variational formulations present the advantage of compactly and elegantly describing a problem via a "well chosen" quantity (or energy) to be optimized. By "well chosen", we mean that the energy is defined so that its (local) minimum is reached when the variables coincide with the solution of a certain problem or with the result of a given task (e.g., segmentation). Typically, one performs optimization (e.g., minimization) of the chosen energy functional by taking its (first) *variation* [48] and using iterative optimization techniques such as gradient descent. Within variational approaches, the dependencies of the diverse variables and parameters can usually be understood simply from the study of the energy functional or its first variation. The explicit aspect of variational techniques facilitates modifications and improvements, in contrast with many alternative approaches that perform image processing tasks by combining different tools and steps, based on heuristics.

Most image processing algorithms can be formulated variationally (see, for instance, [2]). The active contour methodology is an example of variational methods that have been successfully used for performing the afore-mentioned tasks of segmentation and tracking, for example. In the remainder of this section, we present some background information about active contours approaches that will be useful for most chapters of this thesis. The main idea of active contours is to *evolve a curve* C to decrease an energy, and thus perform a certain task. For instance, a curve can be deformed to adopt a particular shape or to delineate the borders of an object of interest in an image (or sequence of images). We now give a formal definition of curves (living in a domain $\Omega \subset \mathbb{R}^2$), which will play a key role in many of the approaches proposed in this work. Let $\mathbf{x} = (x, y)$ denote a point in Ω. A parameterized differentiable curve

C, is a differential map

$$C \ : [a,b] \subset \mathbb{R} \rightarrow \Omega \ ; \ p \mapsto C(p) \doteq [x(p), y(p)]^T = \mathbf{x}(p) \tag{1}$$

In particular, if $C'(p) \neq 0$, for all $p \in [a,b]$, the curve C is said to be "regular". If $C(a) = C(b)$ and $C'(a) = C'(b)$, the curve C is said to be "closed". We will simply refer to "regular closed curves" as "curves", in the reminder of this thesis.

Curve evolution can be defined as a family of curves indexed with an additional (and artificial) time variable t, which is denoted with $C(\cdot, t)$. The family of curves can be completely described by a rule determining how each point on C moves starting from an initial curve C_0, as

$$\frac{\partial C(p,t)}{\partial t} = \mathcal{V}(C(p,t),t), \quad t > 0, \quad s.t. \ C(\cdot, t=0) = C_0 \tag{2}$$

where $\mathcal{V} \ : \mathbb{R}^2 \times \mathbb{R}^+$ is a vector field[1] that specifies the motion of each points of C and is often referred to as the "speed" or "velocity" term. Ignoring the arguments of C and $\mathcal{V}$ in Equation (2) above, one gets

$$\frac{\partial C}{\partial t} = \mathcal{V} \tag{3}$$

The above partial differential equation (PDE) determines how the curve C evolves and is often referred to as "flow". Let $\mathbf{T}(p,t)$ and $\mathbf{N}(p,t)$ denote the unit tangent and (inward) normal[2] to the curve at the point $C(p,t)$, respectively. The two vectors $\mathbf{T}$ and $\mathbf{N}$ form a basis of $\mathbb{R}^2$, in which the speed term can be decomposed as

$$\mathcal{V}(C(p,t),t) = \langle \mathcal{V}(C(p,t),t), \mathbf{T}(p,t) \rangle . \mathbf{T}(p,t) + \langle \mathcal{V}(C(p,t),t), \mathbf{N}(p,t) \rangle . \mathbf{N}(p,t)$$

or, dropping the different arguments, as

$$\mathcal{V} = \langle \mathcal{V}, \mathbf{T} \rangle . \mathbf{T} + \langle \mathcal{V}, \mathbf{N} \rangle . \mathbf{N}.$$

[1] See [33] for a formal definition of vector fields.

[2] The two vectors $\mathbf{T}$ and $\mathbf{N}$ can be expressed as functions of the curve and the chosen parametrization, see for instance [33].

The flow of Equation (3) was shown to be equivalent to the flow

$$\frac{\partial C}{\partial t} = \langle \mathcal{V}, \mathbf{N} \rangle . \mathbf{N} \tag{4}$$

since the tangential term $\langle \mathcal{V}, \mathbf{T} \rangle . \mathbf{T}$ corresponds to a re-parametrization of the curve and does not contribute to its deformation or geometry, see [37] for a formal proof.

An energy (or cost) functional

$$\begin{array}{rccc} E & : \big([a,b] \subset \mathbb{R} \to \Omega\big) & \to & \mathbb{R}^+ \\ & C & \mapsto & E(C) \end{array} \tag{5}$$

expressed in terms of a curve C is said to be "geometric" if it involves only geometric properties of C and is independent of the choice of parametrization. In particular, the gradient of the (geometric) energy E with respect to C and denoted $\nabla_C E$ is a vector field on the curve C that is normal to the curve:

$$\begin{array}{rcl} \nabla_C E(C(p,t),t) & = & \langle \nabla_C E(C(p,t),t), \mathbf{N}(p,t) \rangle . \mathbf{N}(p,t) \\ \nabla_C E & = & \langle \nabla_C E, \mathbf{N} \rangle . \mathbf{N} \end{array} \tag{6}$$

Typically, the energy functional $E(C)$ is minimized via a gradient descent approach, according to the flow

$$\frac{\partial C}{\partial t} = -\nabla_C E = \langle -\nabla_C E, \mathbf{N} \rangle . \mathbf{N} \tag{7}$$

which is conform to the flow of Equation (4), taking $\mathcal{V} = -\nabla_C E$.

To implement the PDE of Equation (3) or (7), one needs to choose a representation for the curve C. Active contours methodologies can be distinguished from the choice of the mathematical representations for curves and classified into two categories: Parametric Active Contours (PAC) (originally proposed in [72, 136]) and Geometric Active Contours (GAC) (originally proposed in [10, 63, 74]). The PAC models employ a Lagrangian formulation in which curves are represented explicitly, typically with splines. Within the PAC methodology, the contour is deformed by updating the corresponding parameters or control points ([72, 20]). The GAC models

use an Eulerian formulation [65] in which curves are represented implicitly. Typically, the contour is embedded in a higher dimensional function using level sets [89]. Within the GAC methodology, the contour is propagated by evolving the embedding function. Level set representations were introduced by Osher and Sethian [89, 116, 87] to model the motion of interfaces and have become the *de facto* standard tools for implementing evolution PDEs, for the following reasons (among others):

- Level sets models can deal with topological changes, such as curve splitting and merging, in a natural fashion (in contrast with parametric models).
- The implicit representation resulting from the use of level sets presents the advantage of avoiding to deal with complex re-sampling schemes of control points.
- The geometric properties of the curve can be directly estimated from the implicit representation.
- The level set representation can be generalized to higher-dimensional objects such as surfaces in $\mathbb{R}^3$, in a straightforward manner.

The main idea of implicit representation consists of representing the contour C by the zero level set of a smooth Lipschitz-continuous function. The most common choice is to embed the contour in a signed distance function (SDF) $\phi : \Omega \to \mathbb{R}$. Let $\omega \subset \Omega$ denote the region located inside the closed curve C, the SDF ϕ can be defined as

$$\phi(\mathbf{x}) = \begin{cases} -\mathcal{D}(\mathbf{x}, C), & \text{for } \mathbf{x} \in \omega; \\ 0, & \text{for } \mathbf{x} \in C; \\ +\mathcal{D}(\mathbf{x}, C), & \text{for } \mathbf{x} \in \Omega \backslash \omega. \end{cases} \tag{8}$$

where

$$\mathcal{D}(\mathbf{x}, C) = \inf_{\mathbf{y} \in C} \mathcal{D}(\mathbf{x}, \mathbf{y})$$

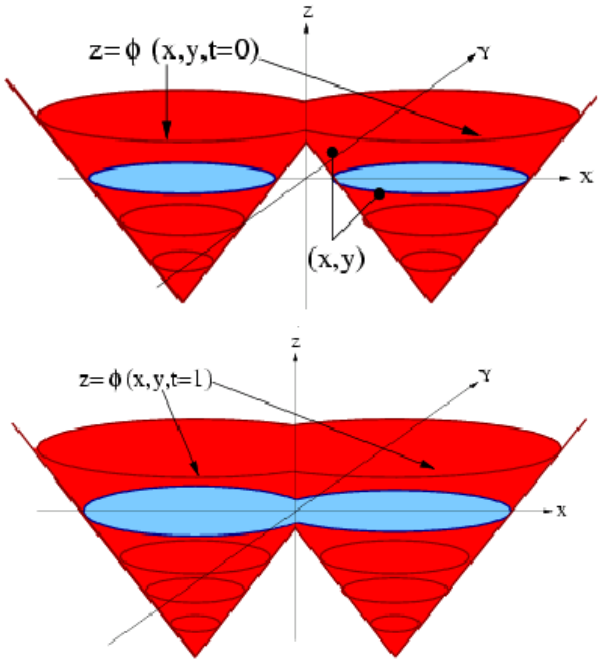


Figure 1: Level Sets of an embedding function ϕ for a closed curve in $\mathbb{R}^2$ (from [116])

denotes the Euclidean distance to the curve C[3]. Hence, and from the definition of ϕ above, one can recover the curve C from the SDF ϕ as

$$C = \{\mathbf{x} = (x, y) \in \Omega \quad s.t. \quad \phi(\mathbf{x}) = \phi(x, y) = 0\} \tag{9}$$

Figure 1.2, presents a visualization of a curve represented implicitly using level sets. The geometric properties of the curve C can be recovered from the representation ϕ. For instance, the inward normal $\mathbf{N}$ can be computed: Noting that $\phi(C(p)) = 0, \ \forall p \in [a, b]$ and differentiating with respect to p, one has

$$\frac{\partial \phi(C(p))}{\partial p} = \langle \nabla_C \phi, \frac{dC}{dp} \rangle = \langle \nabla_C \phi, \| \frac{dC}{dp} \| . \mathbf{T} \rangle = 0.$$

Thus, $\nabla_C \phi$ is orthogonal to $\mathbf{T}$. As a result, one can compute $\mathbf{N}$ from the normalization of $\nabla_C \phi$ as

$$\mathbf{N} = -\frac{\nabla_C \phi}{\|\nabla_C \phi\|} \tag{10}$$

[3]Note that the choice of the sign $(\pm \mathcal{D}(\mathbf{x}, C))$ to distinguish the regions ω and $\Omega \backslash \omega$ in Equation (8) is arbitrary.

where the minus sign comes from the sign convention adopted in the definition of ϕ in Equation (8).

For implementation purposes, the flows of Equation (3) or (7) can be expressed in terms of the level set. From Equation (9) and taking the artificial time parameter t into account, one has

$$\phi(C(p,t),t) = 0.$$

Differentiating the equation above with respect to t and using the chain-rule, one gets

$$\langle \nabla_C \phi, \frac{\partial C}{\partial t} \rangle + \frac{\partial \phi}{\partial t} = 0.$$

Plugging Equation (4) in the equation above yields

$$\frac{\partial \phi}{\partial t} = \|\nabla_C \phi\| . \langle \mathcal{V}, \mathbf{N} \rangle \tag{11}$$

where the fact that $\nabla_C \phi = -\|\nabla_C \phi\| . \mathbf{N}$, from Equation (10), was used (see also e.g., [116]). Equation (11) determines the evolution of the map ϕ, and consequently of C, and is indeed an Eulerian or implicit evolution equation for the curve C (as opposed to the Lagrangian or explicit formulation in Equation (4), for instance). This level set based flow was originally proposed in [89] and further studied in [38, 39, 40, 41], notably. In most cases, the speed term $\mathcal{V}$ (or $\nabla_C E$) is defined only on the curve (or the zero level set) and one needs to extend this definition to all the levels of ϕ for the flow of Equation (11) to make sense. The propagation of the information in the speed term can be carried via "fast marching" methods [1]. Also, fast implementations of the flow in Equation (11) are possible via "narrow-band" approaches that focus on computing quantities only in a neighborhood of the zero level set instead of on the whole domain Ω. Narrow-band techniques were first proposed in [17] and further developed in [74, 1].

In contrast with the definition of energy functional $E(C)$ in Equation (5) that is expressed as a function of the curve C, one can define the energy functional $E(\phi)$

directly in terms of the signed distance function ϕ. Such approach was recently proposed in [12], and have become very popular in the active contour community, see for instance the textbooks [88, 90] and the references therein. The approach relies on smooth approximations of the Heaviside and Dirac delta functions, and often leads to much simplified computations and more efficient implementations (see e.g., Appendixes A and B). Within this methodology, one computes energy gradients directly in terms of the SDF ϕ, instead of in terms of the curve C, as presented above. The resulting gradient descent flows are, thus, of the form

$$\frac{\partial \phi}{\partial t} = -\nabla_{\phi} E \tag{12}$$

The flow equation above also amounts to (implicitly) deforming the contour C from an initial contour C_0 (represented by a SDF ϕ_0) to minimize the energy $E(\phi)$. One should note that flows of Equations (7) and (12) are not strictly equivalent in general. Rather, flows obtained from gradients computed with respect to ϕ can be considered as approximations of flows that result from gradients computed with respect to C. However, the resulting contour evolutions are usually similar notably when narrow-band implementations are used. Because of the efficiency brought about by the approach involving direct computations in terms of ϕ, the flow in Equation (12) will be the default implementation method for the experiments presented in this work, unless otherwise stated.

We conclude this section by noting that one can find interesting further discussions and considerations about the use of signed distance functions to implement curve evolutions in [50, 43, 49, 51].

1.3 Contributions and Organization of this Thesis

Whether they are used to control the shape, position or evolution of the curve, statistical methods are intimately related to the active contour methodology in general and

the GAC framework, in particular. Also, the concept of shape has received a great deal of attention recently, and the use of shape-driven approaches has been shown to often drastically improve the performance of segmentation and tracking algorithms.

The objective of the proposed research is to develop statistical, geometric as well as shape-based tools inspired from or completing the geometric active contours framework. The approaches developed in this thesis make an extensive use of partial differential equations and differential geometry, notably, to address the problems at hand. Most techniques developed in this thesis are cast into a variational framework. This choice was made for the elegance and the rigorous mathematical foundations brought about by this methodology. A particular emphasis will be put on the differences, notably in terms of performance, with currently available techniques, whenever possible. The developed tools are designed to improve the performance of both segmentation and tracking algorithms. Our goal is to develop robust techniques to segment complex images or track difficult video sequences that are corrupted with noise, clutter or occlusion.

This thesis is divided into six main chapters that are meant to be relatively self-contained (except for Chapter 5, which builds on the methods presented in Chapter 3 and Chapter 4). If part of the work has been made publicly available, references to the publications are given in the introduction of the corresponding chapter.

In Chapter 2, a novel tracking algorithm is proposed, in which different statistical and geometric approaches are combined. The algorithm is designed to track the position *as well as* the deformation of the object of interest. The possible shapes of the deformable object are learnt as a pre-processing step of the algorithm. The novelty of the proposed approach is that the coordinates in the learned space of shapes are appended to the coordinates defining the position of the object in the image (the latter are classically used for tracking) to define the state of the system. A robust

(region-based) geometric active contour technique is used as a measurement function to track the object. In addition and in contrast with many other GAC-based tracking schemes, the method maintains the temporal coherency of the state of the object (via a nonlinear filtering technique), while being computationally efficient.

In Chapter 3, a novel segmentation technique with active contours is proposed that uses image information extracted from image regions. Region-based active contour techniques are more robust than methods using only local information. Most region-based active contour techniques focus on separating pixel statistics only, to drive the curve evolution and segment images. The energy functional proposed in this chapter realizes an explicit trade-off between the segmentation obtained from the curve and the segmentation obtained from (dynamically) thresholding the image. The proposed energy has a strong shape interpretation. In contrast with standard region-based techniques, the resulting approach can bypass the need to fit (a priori chosen) statistical models to the object and the background. Also, the technique performs segmentation based on geometric considerations of the image and contour, (instead of or) in addition to purely statistical ones. Hence, the proposed approach exploits more global information in the image, which allows the segmenting curve to reach (local) energy minima that coincide with meaningful segmentations (where standard approaches can fail to do so).

In Chapter 4, diverse methods for including prior knowledge on shapes in the evolution process of geometric active contours are presented. The proposed methods advance the state of the art and are inspired from recent techniques developed in the machine learning community. Within the proposed approaches the possible shapes of objects are learned from sets of training examples. The diverse methods are compared to each other in terms of their accuracy, robustness and ability to learn (and recognize) different objects of various shapes. The superiority of nonlinear (kernel based) learning techniques (over linear ones) is highlighted by numerous examples in

multiple situations. A particular emphasis is made on one nonlinear method (based on kernel principal component analysis) that stands out as a method of choice for introducing shape priors in the geometric active contour framework. In particular, the method is shown to be able to elegantly deal with very complex variations in shapes, to simultaneously learn multiple objects and to constrain the shape of contours in a meaningful and robust manner.

In Chapter 5, a novel framework is proposed that combines statistical information extracted from images with shape information learned a priori from examples. The method takes advantage of the shape interpretation of the energy functionals presented in Chapters 3 and 4 to fuse image information and shape knowledge in a meaningful and homogenous manner. The resulting framework is able to robustly segment challenging images. Furthermore, the method is proven to be able to recognize and choose without supervision the appropriate model (among shape models of diverse objects learned simultaneously) that should be used to perform the segmentation of images featuring a corrupted version of the object of interest.

In Chapter 6, a novel technique is proposed to jointly segment a 3D object of arbitrary shape in a 2D image and estimate its 3D pose with respect to a referential attached to a unique calibrated camera. The proposed method exploits the prior knowledge of the 3D shape of the object of interest. Typically, 2D segmentation with active contours and 3D registration from monocular images are two different tasks studied separately in the literature. The proposed variational approach combines ideas developed in both fields to address some of their typical shortcomings. The resulting framework affords robust performances notably with respect to noise or poor initialization and is quite computationally efficient. One can take advantage of the variational aspect of the technique to perform the tracking of rigid objects that move continuously in a sequence. The method can indeed be used as a measurement function providing not only the position of the object in the image but also in the

world.

In Chapter 7, a novel methodology for the non-deterministic evolution of curves is proposed, based on the theory of interacting particles systems. Curvature driven flows have been extensively considered from a deterministic point of view. A stochastic approximation of curve shortening flows is described that is valid for arbitrary embedded curves on a Riemannian surface. The proposed framework can be used to provide a stochastic interpretation of geodesic active contours, of interest for segmentation tasks in computer vision.

CHAPTER II

TRACKING DEFORMABLE OBJECTS WITH UNSCENTED KALMAN FILTER AND GEOMETRIC ACTIVE CONTOURS

In this chapter, we present an algorithm that combines several statistical and shape-based methods to perform the tracking of deformable objects. This chapter is based on [27] and is organized as follows: In the next section, we provide some motivation and related work to the proposed approach. In Section 2.2, we describe a few state of the art techniques that are used in our algorithm, such as the unscented transformation, the unscented Kalman filter, the Chan and Vese model for region-based segmentation as well as the use of principal component analysis to learn shapes. In Section 2.3, we give a detailed description of the proposed state space model, the measurement model and the algorithm that allow to track both the position and the deformation of moving objects. In Section 2.4, we provide experimental results on challenging real-world sequences.

2.1 Motivation and Related Work

The problem of tracking dynamic deformable objects has been a topic of substantial research in the field of controlled active vision (see [8, 122] and the references therein). In this chapter, we propose a scheme that combines the advantages of the unscented Kalman filter (UKF) and geometric active contours, for dynamic tracking.

To appreciate the proposed methodology, we review some previous related work. Various finite dimensional parameterizations of continuous curves have been proposed,

perhaps most prominently the B-spline representation used for a "snake model", as in [122]. Isard and Blake (see [8] and references therein) applied the B-spline representation for contours of objects and proposed the Condensation algorithm in [57]. The authors in [71, 15] also use B-splines along with the unscented Kalman filter for rigid object tracking. Since these approaches only track the parameters of a finite dimensional group (e.g., Euclidean, affine transformations) they cannot handle severe deformations of the object of interest (see e.g., the fish example in Section 2.4.1). One possible solution proposed in [136], is to use deformable templates to model prior shapes, allowing for a few possible modes of deformation.

Another approach for representing contours is via the level set technique [115, 89]. To segment an object using level sets, an initial guess of the contour is deformed until it minimizes an image-based energy functional. Different energy functionals, which utilize different features of the image have been used in the literature (see e.g., [83, 12, 70, 9, 64, 130]). Some previous work on tracking with GACs is given in [86, 92, 132, 58]. In [132], the authors propose a definition for the global motion and shape deformation of a deformable object. Motion is parameterized by a finite dimensional group action (e.g., Euclidean or affine), while shape deformation is the total deformation of the object contour (infinite dimensional group) modulo the finite dimensional motion group. This is called the *deformotion* model. This approach, as well as most approaches using GAC cited above, rely only on the observed images for tracking and do not use any prior information on the dynamics of the group action or of the deformation. As a result, it is likely to fail if there is an outlier observation or if there is occlusion. To address this problem, the authors in [58] propose a generic local observer to incorporate prior information about the system dynamics in the deformotion framework.

Other approaches closely related to our work are given in [122, 95, 96]. In these papers, the authors use a Kalman filter in conjunction with active contours to track

nonrigid objects. The Kalman filter was used for predicting the possible movements of the object, while the active contours allowed for tracking the deformations of the object.

In [101], the authors use particle filters in combination with geometric active contours for tracking deformable objects. Compared with the approach in [101], our method has the advantage of only requiring a small number of deterministic sample points, and is therefore computationally very efficient. In this work, we combine the advantages of the unscented Kalman filter (e.g., computational efficiency, second order accuracy) and geometric active contours to propose a novel method for tracking deformable objects. The proposed algorithm can maintain the temporal coherency of the state of the system and deal robustly with large deformations, partial occlusions and identity maintenance.

2.2 Preliminaries

2.2.1 The Unscented Transformation

Julier and Uhlmann [61, 60] proposed an approach to generalizing the application of the Kalman filter to nonlinear systems. This approach is based on a statistical technique known as the *unscented transformation.* The unscented transformation leads to a more accurate filter than the traditional extended Kalman filter and avoids the costly computation of Jacobians [61, 128, 127].

Let $\mathbf{x}$ denote an n-dimensional random variable with mean $\hat{\mathbf{x}}$ and covariance $\mathbf{P}$. Let $\mathbf{g}$ be any arbitrary nonlinear function such that $\mathbf{y} = \mathbf{g}(\mathbf{x})$. To calculate the statistics of $\mathbf{y}$, a set of $2n+1$ weighted points or *sigma points* is first deterministically chosen as

$$\chi_0 = \hat{\mathbf{x}}, \quad \omega_0 = \kappa/(n+\kappa) \tag{13}$$

$$\chi_i = \hat{\mathbf{x}} + (\sqrt{(n+\kappa).\mathbf{P}})_i, \quad \chi_{i+n} = \hat{\mathbf{x}} - (\sqrt{(n+\kappa).\mathbf{P}})_i,$$

$$\omega_i = \omega_{i+n} = 1/\{2(n+\kappa)\}, \quad i = 1, ..., n$$

where κ is a scaling parameter, and $(\sqrt{(n+\kappa).\mathbf{P}})_i$ refers to the i^{th} column of the matrix square root of $(n+\kappa) \times \mathbf{P}$. The parameter ω_i is the weight associated with the i^{th} sigma point. Note that $\sum_{i=0}^{2n+1} \omega_i = 1$, and the obtained sigma points have same mean and covariance as $\mathbf{x}$. Each sigma point is then propagated through the nonlinear function $\gamma_i = g(\chi_i)$, with $i = 0, ..., 2n+1$. The estimated mean and covariance of $\mathbf{y}$ are computed as

$$\hat{\mathbf{y}} = \sum_{i=0}^{2n+1} \omega_i \gamma_i, \qquad \mathbf{P}_y = \sum_{i=0}^{2n+1} \omega_i (\gamma_i - \hat{\mathbf{y}})(\gamma_i - \hat{\mathbf{y}})^T$$

2.2.2 The Unscented Kalman Filter

In this section, we briefly present the specificity of the unscented Kalman filter (please refer to [62] for the basics concerning the Kalman filtering approach).

Let $\mathbf{x}$(k) denote the n-dimensional state at time k. The system evolves according to the equation

$$\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{v}(k+1)) \tag{14}$$

where $f(.)$ is the state transition function, and $\mathbf{v}$(k+1) is a q-dimensional process noise vector. The m-dimensional measurement vector $\mathbf{y}$ is linked to the state of the system through the equation

$$\mathbf{y}(k+1) = h(\mathbf{x}(k+1), \mathbf{u}(k+1), \mathbf{w}(k+1)) \tag{15}$$

where $h(.)$ is the observation function in which $\mathbf{u}(k+1)$ is new information available at time $(t+1)$, and $\mathbf{w}$(k) is an r-dimensional measurement noise vector. Although more general assumptions about noise can be carried by the unscented Kalman filter, we assume in what follows that for all integers (i, j)

$$E[\mathbf{v}(k)] = E[\mathbf{w}(k)] = 0 \quad E[\mathbf{v}(i)\mathbf{w}^T(j)] = 0$$

$$E[\mathbf{v}(i)\mathbf{v}^T(j)] = \delta(i,j).\mathbf{Q} \quad E[\mathbf{w}(i)\mathbf{w}^T(j)] = \delta(i,j).\mathbf{R}$$

with $\mathbf{Q}$ and $\mathbf{R}$ being constant matrices of dimensions $(q \times q)$ and $(r \times r)$ respectively. $\mathbf{Q}$ is the process noise covariance matrix and $\mathbf{R}$ is the observation noise covariance matrix. The prediction and update steps of the Kalman filtering algorithm are carried out as described in Section 2.2.2.1 and 2.2.2.2.

2.2.2.1 Prediction

Assume that the mean $\hat{\mathbf{x}}(k)$ and covariance matrix $\mathbf{P}(k|k)$ of the state $\mathbf{x}$(k) at time $t = k$ are known. The unscented transform is applied to the state vector $\mathbf{x}$(k) to obtain a set of $2n + 1$ sigma points $\chi_i(k|k)$ as presented above in Section (2.2.1).

The predicted state can be computed by applying the state transition function $f(\cdot)$ to each of the $\chi_i(k|k)$ to obtain a new set of sigma points $\chi_i(k+1|k)$, defined as

$$\chi_i(k+1|k) = f[\chi_i(k|k)] \quad i = 0, ..., 2n+1 \tag{16}$$

Considering the process noise as additive and independent of the state prediction, the predicted mean and covariance of the state, can be computed as

$$\hat{\mathbf{x}}(k+1|k) = \sum_{i=0}^{2n+1} \omega_i.\chi_i(k+1|k) \tag{17}$$

$$\mathbf{P}(k+1|k) = \sum_{i=0}^{2n+1} \omega_i\{\chi_i(k+1|k) - \hat{\mathbf{x}}(k+1|k)\} \cdot \{\chi_i(k+1|k) - \hat{\mathbf{x}}(k+1|k)\}^T + \mathbf{Q} \tag{18}$$

The predicted observation (measurement) is computed by applying the observation function h(.) to each of the $\chi_i(k+1|k)$ to obtain a new set of sigma points $\gamma_i(k+1|k)$ as

$$\gamma_i(k+1|k) = h[\chi_i(k+1|k), \mathbf{u}(k+1)] \qquad i = 0, ..., 2n+1 \tag{19}$$

Considering the measurement noise as additive and independent of the measurement prediction, the predicted mean and covariance of the measurement, can be computed as

$$\hat{\mathbf{y}}(k+1|k) = \sum_{i=0}^{2n+1} \omega_i.\gamma_i(k+1|k) \tag{20}$$

$$\mathbf{P}_{yy}(k+1|k) = \sum_{i=0}^{2n+1} \omega_i\{\gamma_i(k+1|k) - \hat{\mathbf{y}}(k+1|k)\} \cdot \{\gamma_i(k+1|k) - \hat{\mathbf{y}}(k+1|k)\}^T + \mathbf{R} \tag{21}$$

The predicted cross correlation is given by

$$\mathbf{P}_{xy}(k+1|k) = \sum_{i=0}^{2n+1} \omega_i\{\chi_i(k+1|k) - \hat{\mathbf{x}}(k+1|k)\} \cdot \{\gamma_i(k+1|k) - \hat{\mathbf{y}}(k+1|k)\}^T \tag{22}$$

2.2.2.2 Update

The update step is carried out in the same manner as within the traditional Kalman filter framework. The Kalman gain L, for time $t = k+1$ is

$$L(k+1) = \mathbf{P}_{xy}(k+1|k).\mathbf{P}_{yy}(k+1|k)^{-1}$$

The actual measurement $\mathbf{z}$ can be taken into account, leading to the estimate of the state statistics:

$$\mathbf{x}(k+1) = \hat{\mathbf{x}}(k+1|k) + L.(\mathbf{z} - \hat{\mathbf{y}}) \tag{23}$$

and

$$\mathbf{P}(k+1) = \mathbf{P}(k+1|k) - L.\mathbf{P}_{yy}(k+1|k).L^T \tag{24}$$

2.2.3 The Model of Chan and Vese

Active contours evolving according to edge-based and/or region-based flows are very commonly used for image segmentation (see [110] and the references therein). The level set representation is the most widely used tool for implementing such geometric curve evolution equations (see [87]).

Many methods [107, 123, 70], which incorporate geometric and/or photometric (color, texture, intensity) information have been shown to segment images in the

presence of noise and clutter. In the present work, we have used the Mumford-Shah functional [83] as modeled by Chan and Vese [12] to obtain the curve evolution equation. We seek to minimize the energy functional

$$E_{image} = \nu \int_{\Omega} |\nabla H(\phi)| dxdy + \int_{\Omega} (I - c_1)^2 H(\phi) dxdy + \int_{\Omega} (I - c_2)^2 (1 - H(\phi))\ dxdy \tag{25}$$

where c_1 and c_2 are defined as

$$c_1 = \frac{\int I(x,y) H(\phi) dx\ dy}{\int H(\phi) dx\ dy}, \quad c_2 = \frac{\int I(x,y)(1 - H(\phi)) dx\ dy}{\int (1 - H(\phi)) dx\ dy}$$

and $H(\phi)$ is the Heaviside function defined as

$$H(\phi) = \begin{cases} 1 & if \quad \phi \geq 0\ , \\ 0 & otherwise \end{cases}$$

$I(x, y)$ is the image and ϕ is the level set function corresponding to the segmenting curve. The above energy functional E_{image} can be minimized using calculus of variations. The Euler-Lagrange equation for minimizing this functional can be implemented by the following gradient descent [12, 83]:

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi) \left[\nu\ \text{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - (I - c_1)^2 + (I - c_2)^2 \right] \tag{26}$$

where $\delta_\epsilon(s) = \frac{\epsilon}{\pi(\epsilon^2 + s^2)}$.

2.2.4 Shape Statistics

In [70], the authors apply principal component analysis on a set of signed distance functions to obtain the major modes of shape variation.

Let ϕ_i represent the signed distance function corresponding to the curve C_i. All the ϕ_i's are aligned using a suitable method of registration [18]. The mean surface μ is computed by taking the mean of the signed distance functions:

$$\mu = \frac{1}{n} \sum \phi_i$$

The variance in shape is computed by first subtracting the mean shape μ from each ϕ_i to create a mean-offset map $\bar{\phi}_i$. Then, each such map $\bar{\phi}_i$ is placed as a column vector in an $N^d \times n$-dimensional matrix M, where $\phi_i \in \mathbb{R}^{N^d}$. The covariance matrix can then be computed as

$$C = \frac{1}{n} M M^T$$

Using Singular Value Decomposition (SVD), C can be decomposed as

$$U \Sigma U^T = \frac{1}{n} M M^T \tag{27}$$

where U is a matrix whose column vectors represent the set of orthogonal modes of shape variation, and Σ is a diagonal matrix of corresponding singular values.

Any shape ϕ of the same class of objects (*and registered with the learned shapes*) can be represented by an m-dimensional vector of coefficients, $\alpha = U_m^T(\phi - \mu)$, where U_m is a matrix consisting of the first m columns of U. Given the coefficients α, an estimate $\tilde{\phi}$ of the shape ϕ can be obtained as

$$\tilde{\phi} = U_m \alpha + \mu \tag{28}$$

2.3 Proposed Algorithm

2.3.1 State Space Model

The authors in [132] separate the motion of an object into two distinct parts: a "global" rigid motion, and a "deformation" (any departure from rigidity). They also show that the overall motion of a moving and deforming object can be described by a set of non-unique rigid motion parameters and a deformation function. Accordingly, in this work, we assume that the global motion of an object is given by the translation of its centroid and any other deformation is captured by the curve evolution equation (26) described in the previous section.

We propose to combine the advantages of the unscented Kalman filter and geometric active contours to track both aspects of the motion of an object, position and deformation. A curve being an element of the infinite dimensional space $S^1 \mapsto \mathbb{R} * \mathbb{R}$, a finite dimensional approximation is needed to include the contour into the state space and control it through the unscented Kalman filter. This finite dimensional approximation is obtained using the method presented in 2.2.4, which results in a projection in an orthogonal PCA base (of finite dimension).

In our framework, the state vector is thus composed of the coordinates of the centroid of the object (x_c, y_c) and the curve coordinates in the PCA base (m-dimensional coefficient vector α):

$$\mathbf{x}(k) = \begin{pmatrix} x_c \\ y_c \\ \alpha \end{pmatrix}(k) = \begin{pmatrix} X \\ \alpha \end{pmatrix}(k)$$

Consequently, the dimension of the state vector is $m + 2$. The observation space in this model is also a $m + 2$-dimensional vector given by

$$\mathbf{y}(k) = \begin{pmatrix} x_m \\ y_m \\ \beta \end{pmatrix}(k) = \begin{pmatrix} Y \\ \beta \end{pmatrix}(k)$$

where β is the m-dimensional coefficient vector representing the measured contour and (x_m, y_m) is the measured position. The predicted covariance matrices for the state $\mathbf{x}$ and the measurement $\mathbf{y}$ can be written as

$$\mathbf{P}(k+1|k) = \begin{pmatrix} P & 0 \\ 0 & \Sigma \end{pmatrix}, \ \mathbf{P}_{yy}(k+1|k) = \begin{pmatrix} P_{yy} & 0 \\ 0 & \Sigma_{yy} \end{pmatrix}$$

where P and P_{yy} are 2×2 covariance matrices for the centroid co-ordinates, and Σ and Σ_{yy} are $m \times m$ diagonal matrices obtained as given in (27). Note that we have assumed that the centroid location is independent of the deformation in shape.

The process noise matrix $\mathbf{Q}$ and measurement noise $\mathbf{R}$ are assumed to be constant throughout the state evolution process and are given by

$$\mathbf{Q} = \begin{pmatrix} Q_X & 0 \\ 0 & Q_\Sigma \end{pmatrix}, \mathbf{R} = \begin{pmatrix} R_X & 0 \\ 0 & R_\Sigma \end{pmatrix}$$

2.3.2 Measurement model

The measurement function at time k, $h(\mathbf{X}(k), \mathbf{I}(k))$, where $\mathbf{X}(k)$ is an $(m+2)$-dimensional seed point (corresponding to a curve centered at a certain position), and $\mathbf{I}$(k) is the image that becomes available at time $t = k$, can be described as

1. Build a cloud of l points $\mathbf{X}^{(i)}(k)$, for $i \in [|1..l|]$ around $\mathbf{X}(k)$ (these $(m+2)$-dimensional points correspond to curves centered at certain positions). One way to build this cloud of points is to define a fixed set of $(m+2)$ dimensional vectors $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_l\}$ and to take $\mathbf{X}^{(i)}(k) = \mathbf{X}(k) + \mathbf{s}_i$.

2. Run equation (26), for r iterations for each of the $\mathbf{X}^{(i)}(k)$. The number of iterations r can be chosen according to the expected dynamics of deformation of the object. This results in a local exploration of both the position and shape spaces in the neighborhood of $\mathbf{X}(k)$.

3. Select the curve with the minimum Chan-Vese energy (best fitting curve) as the measurement. The centroid of the selected curve is then taken as a measurement for the centroid of the system. The selected curve is registered to the learned shapes and projected in the PCA base, using equation (28). The projection of the selected curve provides the measurement of the coordinates of the shape.

As can be noticed, the measurement function $h(\mathbf{X}, \mathbf{I})$ is highly nonlinear because of the cumulative effects of the curves evolutions, the selection of the best-fitting

curve and its projection into the PCA base. This nonlinearity of the measurement function justifies the use of the unscented Kalman filter over more classical Kalman filtering approaches in our framework.

2.3.3 Algorithm

Based on the description above, the algorithm can be summarized as

1. Assume the state $\mathbf{x}(k)$ and covariance $\mathbf{P}(k)$ are known, at time $t = k$.

2. Obtain the $2(m+2)+1$ sigma points χ_i as presented in equation (13).

3. Obtain the state predictions using equation (16) to (18).

4. Obtain the measurements predictions using equation (19) to (21), taking $\mathbf{X}(k+1) = \chi_i(k+1|k)$ (for $i = 0, ..., 2(m+2)+1$) as seed points in the definition of the measurement function presented above, which also take $\mathbf{I}$(k+1) as an argument. Typically, using $l = 1$ (and $\mathbf{s}_1 = 0$), in the definition of h above, seemed to provide reasonable estimates for the predicted measurement statistics, for the data tested.

5. Obtain the actual measurement $\mathbf{z}$ by taking $\mathbf{X}(k+1) = \mathbf{x}(k)$ (for example) as a seed point and apply the measurement function on $\mathbf{I}$(k+1). A typical set $\mathbf{S}$ can be chosen to be $\mathbf{S} = \{ \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \end{pmatrix}, \begin{pmatrix} \pm 3 \\ 0 \\ 0 \\ \vdots \end{pmatrix}, \begin{pmatrix} 0 \\ \pm 3 \\ 0 \\ \vdots \end{pmatrix} \}$

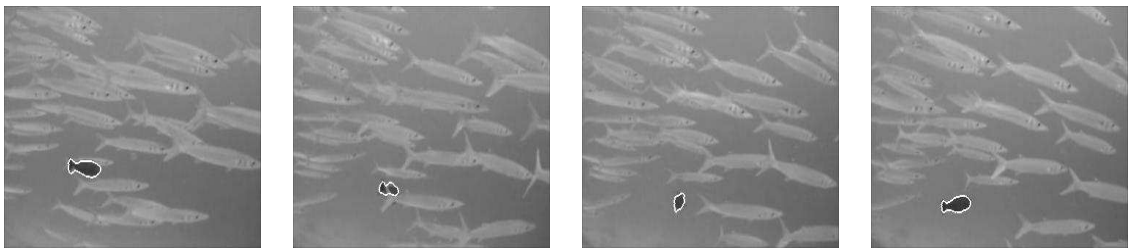
6. Complete unscented Kalman filtering process by computing equations (22) to (24).

2.4 Experiments

In this section, we describe some experiments performed to test the proposed tracking algorithm. Results of applying the proposed method on three image sequences are given below. Geometric active contour implementation was done using the narrow band method [116] and the model of Chan and Vese [12]. In these experiments, the state transition function f in equation (14) was chosen to be identity. For each sequence, a PCA base of the object to track was computed off-line as described before. A small training set of s images was extracted from each sequence and used to compute m components representing the possible modes of variation of the tracked object (eigenvector matrix U_m).

2.4.1 Fish Sequence

The Fish video demonstrates the tracking ability of the proposed method under large deformations and partial occlusions. This type of deformations is difficult to track using the standard Condensation filter [8]. The number of images in the training set and of principal directions was 8, i.e., $s = m = 8$. The total number of images in the sequence was 356. The tracking was accurate throughout the whole sequence, despite the relatively small number of directions of variation allowed for the shape of the fish (Figure 2).

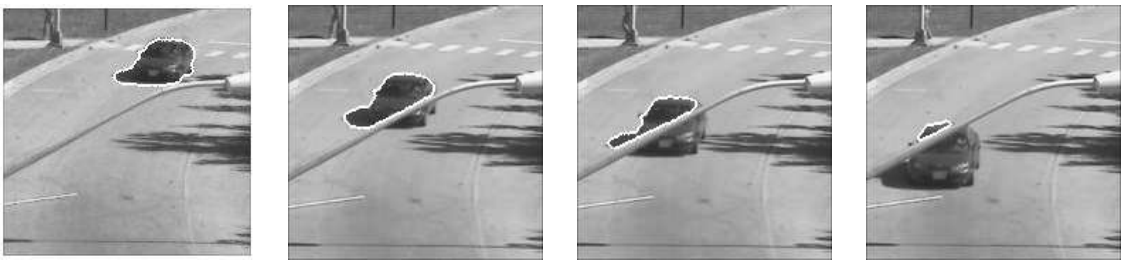


(a) Frame 110 (b) Frame 175 (c) Frame 195 (d) Frame 219

Figure 2: Fish Sequence: Tracking with the proposed method.

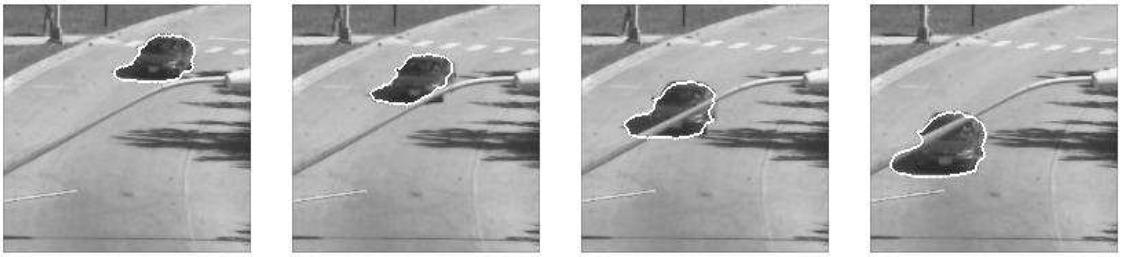
2.4.2 Car Sequence

In this sequence, the car is partially occluded as it passes behind a lamp post. The Chan-Vese model applied to tracking this sequence fails at the occlusion (Figure 3). Including the curve in the state space allows only a few directions of variation in the shape of the car. Hence, the overall shape of the curve does not vary too much from one frame to the next (Figure 4). The results shown on this figure are for $s = m = 6$. Note that even though the vector α provides a shape prior with only 6 possible directions of variation, the proposed model can indeed track large deformations and overcome partial occlusions.



(a) Frame 20 (b) Frame 40 (c) Frame 50 (c) Frame 67

Figure 3: Car Sequence: Tracking with Chan-Vese model alone. the obstacle cannot be handled.



(a) Frame 20 (b) Frame 34 (c) Frame 50 (d) Frame 65

Figure 4: Car Sequence: Tracking with the proposed method.

2.4.3 Walking Couple Sequence

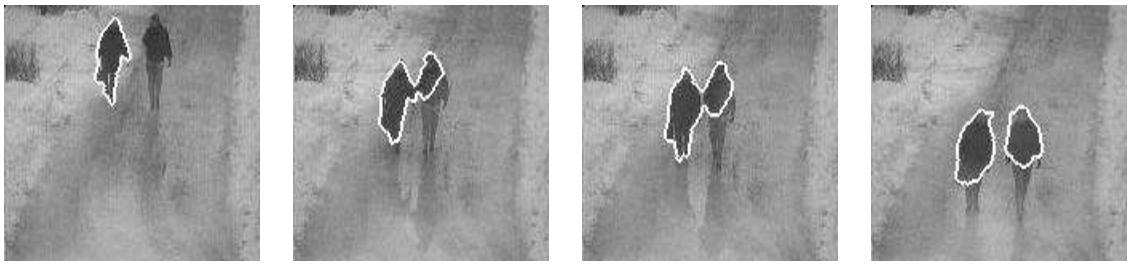
In the classical Walking Couple video, the difficulty resides in maintaining the identity of each person during tracking. Throughout this sequence, the two persons often

touch each other, and the contour usually leaks from one person to the other and ends up encompassing the two people (Figure 5). The identity of each person can be maintained using the proposed method, since leaks are rejected by projecting the contour onto the PCA base (this operation constrains the curve to adopt a shape resembling a walking person). For this sequence, the PCA base was computed from shapes of the person on the right; s and m were 8 and 6, respectively. The person on the left was accurately tracked throughout the sequence, highlighting the robustness of the PCA representation (Figure 6).

2.5 Chapter Conclusion

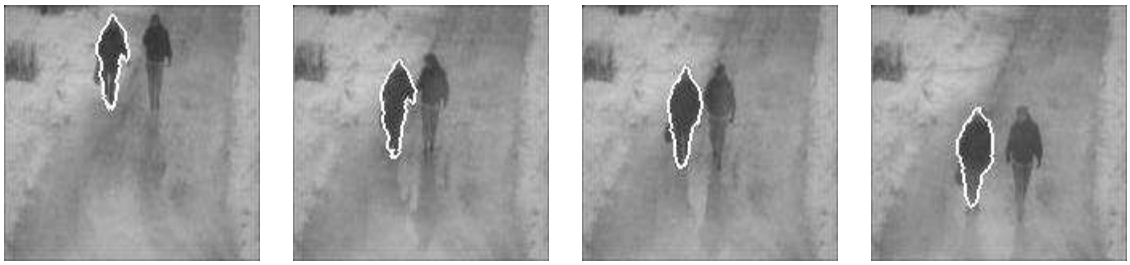
In this chapter, we combined the advantages of the unscented Kalman filter and geometric active contours to propose a method for tracking deformable objects. The proposed method is fast and computationally efficient compared to the one presented in [101], for instance. The algorithm exploits the knowledge of the shape of the object and can deal robustly with large deformations, partial occlusions and identity maintenance.

However, the above framework has limitations, which can be the focus of future work. First, the knowledge of the principal directions of variation is required a priori. Second, shape variation in directions other than the ones represented by U_m can hardly be accommodated. One solution to both limitations could be to devise an online learning approach to update U_m with time. A crucial aspect of this solution would be to determine a collection of criterions that determine over time if and how a new shape should be included (or excluded) in the set of training examples.



(a) Frame 3 (b) Frame 11 (c) Frame 13 (d) Frame 20

Figure 5: Walking Couple Sequence: Tracking with Chan-Vese model. Identity cannot be maintained.



(a) Frame 3 (b) Frame 11 (c) Frame 13 (d) Frame 20

Figure 6: Walking Couple Sequence: Tracking with the proposed method.

CHAPTER III

TAC: THRESHOLDING ACTIVE CONTOURS

In this chapter, we present a novel region-based active contour technique to perform image segmentation. We propose an energy functional that realizes an explicit trade-off between the (current) image segmentation obtained from a curve and the (implied) segmentation obtained from *dynamically* thresholding the image. In contrast with standard region-based techniques, the resulting variational approach can bypass the need to fit (a priori chosen) statistical models to the object and the background. Also, our technique performs segmentation based on geometric considerations of the image and contour, instead of purely statistical ones. This chapter extends preliminary results presented in two international conferences [29, 30] and is organized as follows: In the first section, we provide some motivation and related work to the proposed approach. In the second section, we describe the proposed approach and provide some definitions as well as a description of our energy functional. In the third section, we present the flow resulting from using only geometric information in images. In the fourth section, we present and analyze the flow resulting from using both geometric and statistical information in images. In the two latter sections, experimental results are provided that highlight the specificities of the proposed approach as well as its improved performance over standard techniques, on challenging artificial and real images.

3.1 Motivation and Related Work

Segmentation involves separating an image into distinct regions, a ubiquitous task in computer vision applications. The active contour technique has been proven to be a

very valuable tool for performing this task [12, 21, 72, 138].

Relation to prior work: In the geometric active contour (GAC) framework, a closed curve is represented implicitly as the zero level-set of a higher dimensional function, usually a signed distance function [116, 87]. The curve is evolved to minimize a well-chosen energy functional, typically via gradient descent (see e.g., [116, 90, 87, 88]). The implicit representation allows the curve to naturally undergo topological changes, such as splitting and merging. Different models have been proposed to perform segmentation with GACs: Some frameworks use local image features such as edges [10, 63], whereas other methods use regional image information such as intensity statistics, color or texture [123, 12, 64, 106, 93, 91]. Region-based approaches usually offer a higher level of robustness to noise and a lesser sensitivity to initialization than techniques based on local information. Many of the region-based models have been inspired by the region competition technique proposed in [139]. The book [81] is a nice general reference on the various variational segmentation methods.

In region-based frameworks, intensity statistics are usually estimated from the segmenting curve using parametric [12, 123, 106] or non-parametric [64, 100, 78] methods. Although nonparametric approaches allow one to deal with a wide class of images, parametric methods usually result in simple, robust and efficient segmentation algorithms. Most of the region-based techniques previously proposed *directly* measure discrepancies between the statistics of the pixels located inside and outside the segmenting curve. For example, in [123], the authors increase the L_2-distance between pixel statistics, while in [64] and [100, 78], the authors respectively employ mutual information or the Bhattacharyya correlation measure to separate intensity densities. In [106] (see also [12, 119]), the log-likelihood is used to distinguish distributions.

Motivation: However, segmentation can be regarded as the problem of finding a certain shape in an image and the statistics of pixels intensities are only a means

to this end. Based upon this observation, we propose a region-based segmentation technique using GACs. We present a variational formulation, in which we minimize an energy functional that compares two shapes: The shape of the segmenting contour and a shape extracted from the image via a thresholding operation. Our methodology does not require to fit (a priori chosen) statistical models of regions. However, pixels statistics may be used (in an indirect fashion) in the proposed methodology. The flow resulting from the definition of our new energy exhibits distinctive properties that stem from our shape-based approach to the segmentation problem and lead to meaningful segmentations, in comparison with standard region-based algorithms.

3.2 Proposed Approach

3.2.1 Notation and Problem Formulation

We consider the problem of segmenting an image $I : \Omega \mapsto \mathcal{Z}$, where $\Omega \subset \mathbb{R}^2$ is the image domain and $\mathcal{Z} \subset \mathbb{R}^n$ is the space of pixel intensity values [1]. The area element of Ω will be denoted $d\Omega$ and $x \in \Omega$ will specify the coordinates of the pixels in the image I. We assume that I is composed of two (unknown) regions, referred to as "Object" and "Background." The goal of segmentation is to capture these two regions.

To do so, we evolve a closed curve C, represented as the zero level-set of a signed distance function $\phi : \Omega \mapsto \mathbb{R}$, such that $\phi > 0$ inside C and $\phi < 0$ outside C. Our goal is to evolve the contour C, or equivalently ϕ, so that its interior matches the Object, and its exterior matches the Background: the curve C would then match the boundary $\partial\Omega$ separating Object and Background. The region inside C (respectively outside) will be denoted R (respectively $R^c = \Omega \setminus R$).

[1] For instance, $n = 1$ in the case of gray-scale images and $n = 3$ in the case of color images

Let us denote by $H : \mathbb{R} \mapsto \{0, 1\}$ the Heaviside step function,

$$H(\chi) = H\chi = \begin{cases} 1 & \text{if } \chi > 0; \\ \frac{1}{2} & \text{if } \chi = 0; \\ 0 & \text{if } \chi < 0. \end{cases}$$

The derivative of H will be denoted by $\delta(\chi) = \delta\chi = \frac{dH}{d\chi}$.

In this chapter, a shape $\mathcal{S} \subset \Omega$ is characterized by a labeling function $\mathrm{S}_{\mathcal{S}} : \Omega \mapsto \{0, 1\}$ such as

$$\mathrm{S}_{\mathcal{S}}(x) \begin{cases} = 1 & \text{if } x \in \mathcal{S}; \\ = 0 & \text{if } x \in \Omega \backslash \mathcal{S}. \end{cases}$$

The function H described above can be used as a labeling function to describe shapes. For instance, the shape defined by the interior of the contour C can be characterized by the labeling function $H\phi$.

Let $G(\tau, I(x)) = 1 \quad \text{or} \quad 0$ denote the result of thresholding the image using the threshold(s) $\tau = (\tau_1, ..., \tau_N)$ and with the convention that pixels such that $G(\tau, I(x)) = 1$ correspond to pixels deemed to belong to the Object (see e.g., [53, 117], for more details about image thresholding). From the definition of shape above, the functional G can be interpreted as a shape extracted from the image.

3.2.2 Energy Functional

In most of the region-based segmentation techniques, the assumption is made that the Object and Background are characterized by statistical properties, which are visually consistent and distinct from each other. From this hypothesis, most of the region-based techniques that do not assume prior knowledge of intensity distributions, work directly with the densities P_{in} and P_{out} (the conditional distributions of the pixels inside and outside the curve, respectively) to increase their difference in the hope to

converge towards the densities P_{O} and P_{B} (the distributions of the pixels belonging to the Object and the Background respectively). Typically, an energy functional that measures the similarity between P_{in} and P_{out} (or simply statistical moments) is defined and minimized. Thus, the differences among approaches mainly resides in the choice of the measure of similarity among intensity statistics or densities.

In this work, we propose to minimize the following energy, to perform segmentation

$$E_{\mathrm{image}}(\phi, \tau) = \|H\phi - G(\tau)\|^2 = \int_{\Omega} \big(H\phi(x) - G\left(\tau, I(x)\right)\big)^2 d\Omega \tag{29}$$

In this expression the energy is a function of the level-set ϕ. The energy E_{image} can be equivalently re-written as a functional involving the curve C, through R and R^c as

$$E_{\mathrm{image}}(C, \tau) = \int_{\mathrm{R}} \big(1 - G\left(\tau, I(x)\right)\big) d\Omega + \int_{\mathrm{R^c}} G\left(\tau, I(x)\right) d\Omega \tag{30}$$

In contrast with standard region-based approaches, this energy does not necessarily involve pixel statistics inside and outside the curve. Minimizing such an energy can be beneficial when statistical models cannot properly distinguish between regions of interest. Furthermore, one can note that regions in two images can have rigourously the same statistics but represent very different objects. As a result, segmentation can be regarded as the problem of finding a certain shape in an image, and the statistics of pixels intensities are only a means to this end. Thus, focusing on statistics alone can be myopic.

As can be noted from Equation (29), the energy E_{image} is the L_2 distance between two shapes: the shape of the region defined by the interior of the curve C and the shape G extracted from the image. This is in accordance with the intuition that segmentation is about extracting shapes, and the construction of the functional G is acquainted to learning the shape of the object of interest online. Furthermore, minimizing E_{image} effectively amounts to realizing a trade-off between the segmentation obtained from the curve C and the segmentation obtained from thresholding the

image (N.B.: The thresholding operation is taken *as a function of the curve* C in this work. In particular, G varies when C varies).

This set-up allows us to take advantage of the flexibility of the thresholding technique (see e.g., [117]), while palliating some of its typical weaknesses such as lack of locality (thresholding is a global operation, and supervision is usually required to choose parts of interest in the thresholded image). Here, the locality of the active contour is exploited. Also, meaningful geometric constraints can be imposed to the curve C to improve segmentation results. For instance, the smoothness of the results can be adjusted using a curvature term (such regularization is difficult to impose on thresholding approaches)

$$E_T(C, \tau) = E_{\text{image}}(C, \tau) + \int_C ds. \tag{31}$$

Obviously, many choices are possible for the thresholding function G. In what follows, we study the flows minimizing E_{image} that are obtained for diverse choices of G. In the first type of flow, the function G is built using optimal thresholds for the energy E_{image}. This allows one to exploit *only* geometric information in the image. In the second type of flow, the function G is built using thresholds computed from image statistics inside and outside the segmenting curve. This allows one to exploit *both* geometric as well as statistical information.

3.3 Pure Geometric Case

3.3.1 Gradient Flow

In Equation (30), the unknowns are the curve C and the threshold(s) τ. We now present the details of the flow obtained when one solves optimally for both C^* and τ^* for grey-scale images ($\mathcal{Z} = [0..255]$). In this case, one has a joint minimization problem of the form

$$(C^*, \tau^*) = \arg\min_{C,\tau} \left(E_{\text{image}}(C, \tau)\right) \tag{32}$$

Hence, in this case where optimal threshold(s) are computed, the energy E_{image} compares the two shapes $H\phi$ and G based only on geometric considerations: it is minimal when the *graphs* of the label maps $H\phi$ and G coincide in a maximal fashion over the entire domain Ω.

3.3.1.1 Thresholding with a unique threshold

The elementary way of thresholding the grey-scale image I with the threshold τ is to compute

$$G(\tau, x) = H\big(I(x) - \tau\big) \tag{33}$$

when the object of interest is lighter than the background, or

$$G(\tau, x) = 1 - H\big(I(x) - \tau\big) = H\big(-(I(x) - \tau)\big) \tag{34}$$

when the object of interest is darker than the background.

For the definition of G in Equation (33), one can compute $E'_{\text{image}}(\tau) = \frac{\partial E_{\text{image}}}{\partial \tau}$ from Equation (30)

$$\frac{\partial E_{\text{image}}}{\partial \tau} = \int_R \delta\big(I(x) - \tau\big) - \int_{R^c} \delta\big(I(x) - \tau\big) \tag{35}$$

Using the scaling property of the Dirac delta function generalized to multidimensional functionals (co-area formula), one gets

$$\frac{\partial E_{\text{image}}}{\partial \tau} = \int_{\Gamma \cap \mathrm{R}} \frac{ds'}{\|\nabla_x I\|} - \int_{\Gamma \cap \mathrm{R^c}} \frac{ds'}{\|\nabla_x I\|} \tag{36}$$

where $\nabla_x I$ denotes the gradient of the image with respect to the spatial (Euclidean) coordinates. The curve Γ, parameterized with arclength s', represents the isolines in I such that $I = \tau$, i.e.:

$$\Gamma = \{x \in \Omega \quad \text{s.t.} \quad I(x) = \tau\}$$

Hence, the curve $\Gamma = \partial G$ corresponds to the edges of the thresholded image G. One notes that Equations (35) and (36) are defined if and only if $\nabla_x I \neq 0$ on Γ - This

can be easily enforced on quantized images, as described in the sequel, ensuring that $E'_{\text{image}}(\tau)$ is well defined.

Considering the function $E'_{\text{image}}(\tau)$ in Equations (35) or (36), one can compute the *optimal* threshold $\tau^*(C)$ for the energy E_{image}, given the curve C. This optimal threshold is such that $\frac{\partial E_{\text{image}}}{\partial \tau}(\tau^*(C)) = 0$, $\frac{\partial E_{\text{image}}}{\partial \tau}(\tau^{*+}(C)) > 0$, and $\frac{\partial E_{\text{image}}}{\partial \tau}(\tau^{*-}(C)) < 0$.

Once the threshold $\tau^*(C)$ is computed, the energy E_{image} can be minimized with respect to C using gradient descent, with

$$\nabla_C E_{\text{image}}(x) = \big(1 - 2G(\tau^*(C), x)\big).\mathbf{N}(x) \tag{37}$$

where $\mathbf{N}(\mathbf{x})$ is the outward normal to the curve at the point x. The flow minimizing the Energy E_T, is then

$$\frac{dC}{dt} = -\big(1 - 2G\big).\mathbf{N} - \kappa.\mathbf{N} \tag{38}$$

where $\kappa(x)$ is the curvature of the curve C at x.

One then iterates the process described above, alternating between the computation of $\tau^*(C)$ and the evolution of C, until the convergence of both $\tau^*(C)$ and C to τ^* and C^*, respectively.

In practice, and from an initial guess ${\tau^*_{t=0}}^2$ for τ^*, one can detect changes in the sign of $E'_{\text{image}}(\tau) = \frac{\partial E_{\text{image}}}{\partial \tau}$, for quantized values of the thresholds τ (i.e., values of $\tau \in \mathcal{Z}_{\frac{1}{2}} = \{-0.5, 0.5, 1.5, ...254.5\}$). Let τ_a and $\tau_b = \tau_a + 1$ be the threshold values, closest to $\tau^*_{t=0}$, such that $E'_{\text{image}}(\tau)$ changes sign. One can then choose $\tau^*(C)$ to be the threshold between τ_a and τ_b that leads to the lowest energy E_{image}.

$$\tau^*(C) = \arg\min_{\tau_a, \tau_b} E_{\text{image}}(C, \tau) \tag{39}$$

[2]Typically one can take the average of the mean intensities inside and outside C for $\tau^*_{t=0}$, at the very first step of the contour evolution. For successive steps, the value $\tau^*_{t=0}$ is taken to the value τ^* found in the preceding step.

N.B. 1: The computations of τ_a, τ_b and $\tau^*(C)$ above are valid for G computed using Equation (33). However, one can note that the expression of $E'_{\text{image}}(\tau)$ simply changes sign when G is computed using Equation (34). Hence, one can *first* perform the computations of τ_a, τ_b and $\tau^*(C)$ using the expression of $E'_{\text{image}}(\tau)$ in Equation (35), and *then* compute G using Equation (33) if $E'_{\text{image}}(\tau_a) < 0$ and $E'_{\text{image}}(\tau_b) > 0$ or using Equation (34) if $E'_{\text{image}}(\tau_a) > 0$ and $E'_{\text{image}}(\tau_b) < 0$.

N.B. 2: The derivative $E'_{\text{image}}(\tau)$ is well defined, for the thresholds $\tau \in \mathcal{Z}_{\frac{1}{2}}$ since $I(x) \in \mathcal{Z}$ and, thus, necessarily $\nabla_x I \neq 0,\ \forall x \in \Gamma$

3.3.1.2 Thresholding with multiple thresholds

Thresholding the grey-scale image I with multiple thresholds $\tau = (\tau_1, ..., \tau_N)$ can be undertaken by computing

$$G(\tau, x) = H\left(\pm \prod_{i=1}^{N} \big(I(x) - \tau_i\big)\right) = H\left(\mathcal{T}_m(x, \tau)\right) \tag{40}$$

Let us compute the derivative of E_{image} with respect to a threshold τ_{i0}

$$\begin{aligned} \frac{dE_{\text{image}}}{d\tau_{i0}} = &\int_R \delta\big(\mathcal{T}_m(x,\tau)\big) \left(\pm \prod_{i \neq i0} \big(I(x) - \tau_i\big)\right) d\Omega \\ &- \int_{R^c} \delta\big(\mathcal{T}_m(x,\tau)\big) \left(\pm \prod_{i \neq i0} \big(I(x) - \tau_i\big)\right) d\Omega \end{aligned} \tag{41}$$

Similarly to the section above, we define the curve Γ_{i0}, parameterized with arclength s' as

$$\Gamma_{i0} = \{x \in \Omega \quad \text{s.t.} \quad I(x) = \tau_{i0}\}$$

Also, one can compute the gradient of $\mathcal{T}_m$ with respect to spatial coordinates as

$$\nabla_x \mathcal{T}_m = \pm \sum_{k=1}^{N} \left(\prod_{i \neq k} \big(I(x) - \tau_i\big)\right) \nabla_x I \tag{42}$$

For $x \in \Gamma_{i0}$, one has

$$\nabla_x \mathcal{T}_m(x) = \pm \prod_{i \neq i0} \big(\tau_{i0} - \tau_i\big) \nabla_x I \tag{43}$$

Assuming again that $\nabla_x I \neq 0$ and using the scaling property of the Dirac delta function generalized to multidimensional functionals, one gets

$$\frac{dE_{\text{image}}}{d\tau_{i0}} = \text{sign}\left(\pm \prod_{i \neq i0} (\tau_{i0} - \tau_i)\right) \left\{ \int_{R \cap \Gamma_{i0}} \frac{ds'}{\|\nabla_x I\|} - \int_{R^c \cap \Gamma_{i0}} \frac{ds'}{\|\nabla_x I\|} \right\} \tag{44}$$

Considering the function

$$\beta(\tau_{i0}, C) = \left\{ \int_{R \cap \Gamma_{i0}} \frac{ds'}{\|\nabla_x I\|} - \int_{R^c \cap \Gamma_{i0}} \frac{ds'}{\|\nabla_x I\|} \right\}$$

of the variable τ_{i0} (for a given curve C), one can detect the values of τ_{i0} for which the derivative $\frac{dE_{\text{image}}}{d\tau_{i0}}(\tau_{i0}) = 0$. In practice, this can be done similarly as in the section above by detecting the sign changes of $\beta(\tau_{i0})$, for quantized values of the thresholds τ_{i0} (e.g., values of $\tau_{i0} \in \mathcal{Z}_{\frac{1}{2}}$). Let N be the number of sign changes detected. Let $\tau_{a,j}$ and $\tau_{b,j} = \tau_{a,j} + 1$ (for $j \in [1, N]$) be quantized values such that $\beta(\tau_{i0})$ changes sign (i.e., $\beta(\tau_{a,j})\beta(\tau_{b,j}) < 0$). For all $j \in [1, N]$, one can define

$$\tau_j^*(C) = \arg \min_{\tau_{i0} = \tau_{a,j}, \tau_{b,j}} \left(|\beta(\tau_{i0})| \right) \tag{45}$$

One can compute the thresholded image $G^*(C)$, using $\tau = \tau^*(C) = (\tau_1^*(C), ..., \tau_N^*(C))$ in Equation (40) (Choosing the sign of the argument of the Heaviside function that leads to the minimum of the energy E_{image}). This thresholded image $G^*(C)$ is then the *optimal thresholded image* corresponding to the *global* minimum of E_{image}, given C (this can be proven studying the function $E_{\text{image}}(\tau_{i0})$ and using the definition of the thresholds $\tau_j^*(C)$).

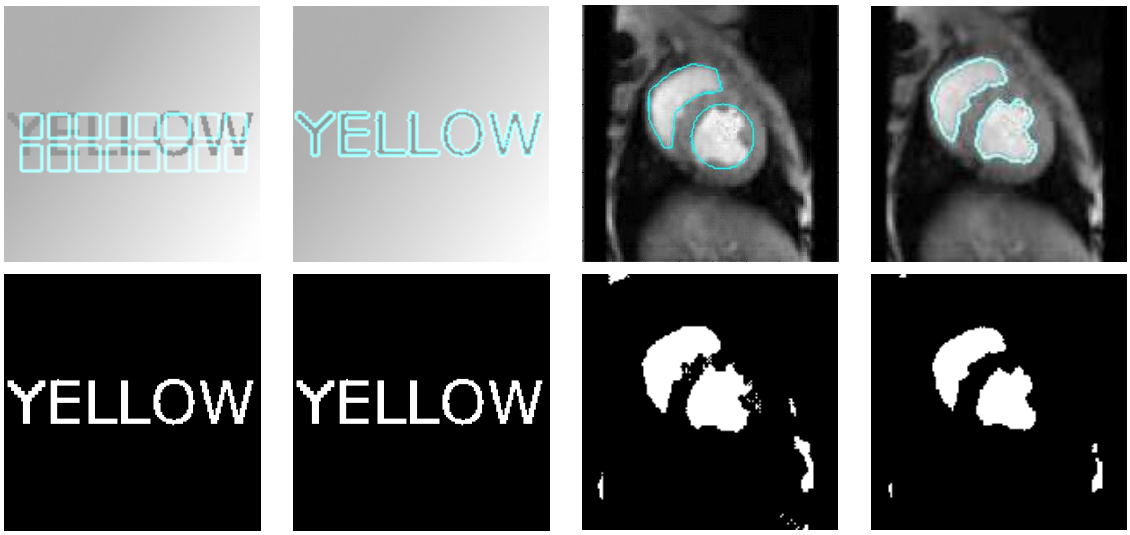
One can note that the procedure described above to compute G^* allows not only to compute the optimal thresholds $\tau_j^*(C)$ but their number N. Knowing the number of thresholds to use a priori is a challenging problem for most standard thresholding techniques. Here, the information embedded in the shape of the curve C is used to determine the optimal number of thresholds. This is an interesting feature of the proposed method.

3.3.2 Experiments for the Purely Geometric Case

We now present experiments that illustrate the performance of the proposed method, using optimal thresholds (i.e.: using geometric information only). Figure 7 presents segmentation results on a synthetic image and an MRI image of a heart. The proposed method was used on both image using a unique threshold as presented in Section 3.3.1.1. The word "Yellow" is accurately segmented using our technique in the artificial image. Also, the left ventricle is accurately segmented in the MRI image (the corresponding thresholded images G are presented. Note that these models can *vary* during the contour evolution). In Figure 8, two purely statistical methods ([12] and [106]) are run on the same two images for identical initializations. The two methods fit Gaussian models to the regions inside and outside the curve to perform segmentation. For both images, the two methods fail to lead to satisfying segmentations and end up capturing much bigger regions than the regions of interest (even if the contours are initialized very close to the ventricles for instance). These two images are typical cases where statistical models do not properly distinguish between regions of interest and where the geometric approach leads to superior results.

In Figure 9, an image of a stuffed monkey under complex lighting conditions is successfully segmented by the proposed method using multiple thresholds. At initialization, 6 optimal thresholds are found, whereas 5 thresholds are used around convergence.

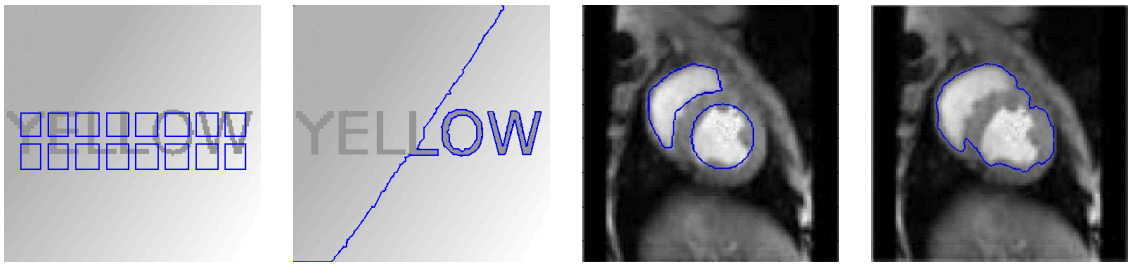
In Figure 10, the MRI image is segmented again using the proposed approach (using a unique and multiple optimal thresholds) with a different initialization. A poor segmentation result is obtained that captures only parts of the left ventricles. This illustrates the sensitivity to initialization of the method when using geometric considerations only (with the initialization of Figure 7, a satisfying segmentation was obtained). Using the methods in [12, 106] based on statistical considerations only,



(a) (b) (c) (d)

Figure 7: Segmentation of an artificial and an MRI image with the purely geometric method, using a unique optimal threshold. *Top line:* (a): Initialization for the artificial image; (b): Result; (c): Initialization for the MRI image; (d): Result. *Bottom line :* Tresholded images corresponding to the contours in the top line.

the *same* (unsatisfying) result as in Figure 7 is obtained were the contour captures more than the region of interest. Hence, the purely geometric approach tends to over-segment images (i.e. capture smaller regions than the regions of interest) and be quite sensitive to initialization. Conversely, purely statistical approaches tend to under-segment images (i.e. capture bigger regions than the regions of interest) but appear to be less sensitive to initialization. Combining both geometric and statistical methodologies can, thus, be expected to lead to more robust segmentation performances (more accurate segmentation due to a trade off between over and under segmentation as well as robustness to initialization). This is the subject matter of the following section.



(a) (b) (c) (d)

Figure 8: Segmentation of an artificial and an MRI image with the methods in [12, 106] that are based purely on statistical considerations. *Top line:* (a): Initialization for the artificial image; (b): Result; (c): Initialization for the MRI image; (d): Result.
.

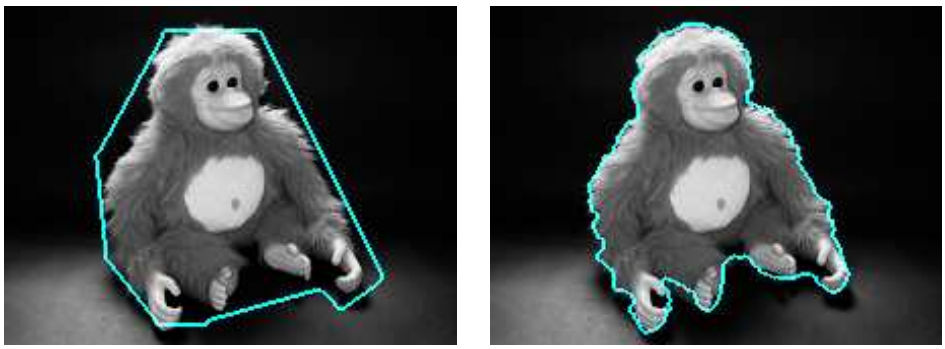


Figure 9: Segmentation of an image under complex lighting conditions, using multiple optimal thresholds. *Left:* Initialization - 6 optimal thresholds used; *Right:* Result - 5 optimal thresholds used.

3.4 Joint Statistical and Geometric Case

3.4.1 Gradient Flow

According to standard region-based techniques we now use statistical models for P_{in} and P_{out} in the proposed framework. These densities P_{in} and P_{out} can be estimated/computed at each step of the contour evolution and can be expressed as functionals of C or ϕ (as will be detailed below). Given a meaningful initialization of the evolving curve, these statistics provide valuable information about the statistics of the Object and the Background. Intuitively, the conditional densities P_{in} and P_{out} are respectively the best available estimates of the *unknown* conditional densities P_{O} (density of the pixel intensities belonging to the Object) and P_{B} (density of the pixel

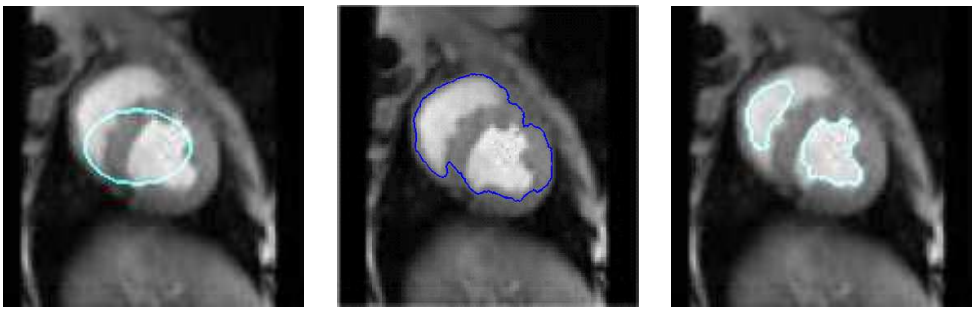


Figure 10: Comparative segmentation of an MRI image for an initialization quite far from the region of interest. *Left:* Initialization; *Middle:* Typical result with the methods in [12, 106] that are based purely on statistical considerations; *Right:* Result with the purely geometric approach proposed (with a unique as well as multiple optimal thresholds)

intensities belonging to the Background).

Instead of computing *optimal* thresholds to reduce the energy E_{image} as in Section 3.3, we now define the thresholding function $G : \Omega \mapsto [0, 1]$ so that pixels x that are more likely to belong to the Object, i.e., $P_{\text{in}}\big(I(x)\big) \geq P_{\text{out}}\big(I(x)\big)$, are assigned the value $G(x) = 1$, whereas pixels more likely to belong to the Background, i.e., $P_{\text{in}}\big(I(x)\big) < P_{\text{out}}\big(I(x)\big)$, are assigned a value $G(x) = 0$. Using the log-likelihood as a discrepancy measure of distributions[3], a candidate for the function G is

$$G(x) = H\big[\log\big(P_{\text{in}}(I(x))\big) - \log\big(P_{\text{out}}(I(x))\big)\big] \tag{46}$$

The threshold(s) corresponding to the definition of G are not in general optimal for the energy E_{image}, but rather are *constrained* upon the choice of the statistical models for P_{in} and P_{out}. Intuitively and from our definition of shape above, the functional G can be interpreted as the most likely shape of the Object, based on the knowledge of the intensity distributions of pixels inside and outside the curve. Hence, the label map G is a functional of the curve C (and of its representation ϕ).

[3]Other measures of discrepancy between pixels distributions of regions could be used to perform the thresholding operation, see e.g. [117] and the references therein. This particular choice of the log-likelihood was made here to allow to compare the proposed approach with standard region-based active contour techniques such as [12, 93, 106] that use the same discrepancy measure, in a meaningful fashion.

One can note that the energy E_{image} now exploits even more global information in the image than standard region-based technique. Indeed, two types of global information are measured through the energy E_{image}: the discrepancy of the global pixel statistics inside and outside the curve (which is classical) and the accordance of the geometric properties of the maps $H\phi$ and G (which is a novel contribution of this work).

3.4.1.1 Thresholding with a unique explicit threshold

In [12], a method is proposed to segment *greyscale* images composed of regions of different mean intensities, using GACs. The resulting flow can be shown to be equivalent to comparing the log-likelyhood of the Gaussian densities

$$\begin{aligned} P_{\text{in}}(I,C) &= \mathcal{N}_{\mu_{\text{in}},\Sigma_0}(I,C) = \frac{1}{\sqrt{2\pi\Sigma_0}} e^{-\frac{(I-\mu_{\text{in}})^2}{2\Sigma_0}} \quad \text{and} \\ P_{\text{out}}(I,C) &= \mathcal{N}_{\mu_{\text{out}},\Sigma_0}(I,C) = \frac{1}{\sqrt{2\pi\Sigma_0}} e^{-\frac{(I-\mu_{\text{out}})^2}{2\Sigma_0}}. \end{aligned} \tag{47}$$

where the intensity averages of the pixels located inside and outside the curve C are denoted by μ_{in} and μ_{out} respectively, and $\Sigma_0 = \frac{1}{2}$. The averages μ_{in} and μ_{out} are computed at each step of the curve evolution as

$$\mu_{\text{in}}(C) = \frac{\int_R I(x)\, d\Omega}{A_{\text{in}}} \text{ and } \mu_{\text{out}}(C) = \frac{\int_{R^c} I(x)\, d\Omega}{A_{\text{out}}} \tag{48}$$

with $A_{\text{in}}(C) = \int_R d\Omega$ and $A_{\text{out}}(C) = \int_{R^c} d\Omega$, the areas inside and outside the curve, respectively. The flow in [12] (Chan-Vese flow) is

$$\begin{aligned} \text{Chan} - \text{Vese flow}: \quad \frac{dC}{dt} &= \left(\log\left(P_{\text{in}}\right) - \log\left(P_{\text{out}}\right)\right).\mathbf{N} = ((I-\mu_{\text{out}})^2 - (I-\mu_{\text{in}})^2).\mathbf{N} \\ &= ((\mu_{\text{out}}-\mu_{\text{in}})(\mu_{\text{out}}+\mu_{\text{in}}) + 2I(x)(\mu_{\text{in}}-\mu_{\text{out}})).\mathbf{N} \\ &= (\mu_{\text{in}}-\mu_{\text{out}})(-\mu_{\text{out}}-\mu_{\text{in}}+2I(x)).\mathbf{N} \\ &= 2(\mu_{\text{in}}-\mu_{\text{out}})(I(x)-\tau_0).\mathbf{N} \end{aligned} \tag{49}$$

where the threshold τ_0 is defined as

$$\tau_0(C) = \frac{\mu_{\text{in}} + \mu_{\text{out}}}{2} \tag{50}$$

Hence, according to the flow in Equation (49), the curve moves inward or outward depending on the sign of the function of the threshold $\mathcal{T}_{CV}(x) = (\mu_{\text{in}} - \mu_{\text{out}})(I(x) - \tau_0)$.

Accordingly and using the definition of G in Equation (46), we define

$$\begin{aligned} G(x) &= H\left(\log\left(P_{\text{in}}(I(x))\right) - \log\left(P_{\text{out}}(I(x))\right)\right) \\ &= H[(\mu_{\text{in}} - \mu_{\text{out}})(I(x) - \tau_0)] = H[\mathcal{T}_{CV}(x)] \end{aligned} \tag{51}$$

where the scaling-property of the Heaviside function $H(\alpha\chi) = H(\chi),\ \forall\alpha \in \mathbb{R}^+$, was used. Contrary to Section 3.3.1.1, one has an explicit (closed-form) formula to compute the threshold τ_0 at each step of the contour evolution from Equation (50). One can note that the factor $(\mu_{\text{in}} - \mu_{\text{out}})$ in the expression of τ_0, compactly distinguishes the cases when the object is either lighter or darker than the background, similarly to the distinction made in Section 3.3.1.1.

The threshold τ_0 is not in general the optimal threshold for the energy E_{image} given the contour C. Using the definition of τ_0 above when minimizing E_{image} with respect to C, results in a constrained optimization of the form

$$\begin{cases} C^{**} = \arg\min_C \left(E_{\text{image}}(C, \tau)\right) \\ \tau(C) = \tau_0 \end{cases} \tag{52}$$

Similarly to the optimization presented in the sections above, the (constrained) minimization of E_{image} is carried via gradient descent. The Gradient of E_{image} with respect

to the curve C can be computed, using Equation (30), as

$$\begin{aligned}\nabla_C E_{\text{image}} &= (1-2G).\mathbf{N} \\ &+ \frac{1}{2}\left\{\int_R \delta[(\mu_{\text{in}}-\mu_{\text{out}})(I(x)-\tau_0)].(\mu_{\text{in}}-\mu_{\text{out}})d\Omega\right\}(\nabla_C\mu_{\text{in}}+\nabla_C\mu_{\text{out}}) \\ &- \left\{\int_R \delta[(\mu_{\text{in}}-\mu_{\text{out}})(I(x)-\tau_0)].(I(x)-\tau_0)d\Omega\right\}(\nabla_C\mu_{\text{in}}-\nabla_C\mu_{\text{out}}) \\ &- \frac{1}{2}\left\{\int_{R^c} \delta[(\mu_{\text{in}}-\mu_{\text{out}})(I(x)-\tau_0)].(\mu_{\text{in}}-\mu_{\text{out}})d\Omega\right\}(\nabla_C\mu_{\text{in}}+\nabla_C\mu_{\text{out}}) \\ &+ \left\{\int_{R^c} \delta[(\mu_{\text{in}}-\mu_{\text{out}})(I(x)-\tau_0)].(I(x)-\tau_0)d\Omega\right\}(\nabla_C\mu_{\text{in}}-\nabla_C\mu_{\text{out}})\end{aligned} \tag{53}$$

Regrouping similar terms, one has

$$\begin{aligned}\nabla_C E_{\text{image}} &= (1-2G).\mathbf{N} \\ &+ \left\{\int_R \delta[(\mu_{\text{in}}-\mu_{\text{out}})(I(x)-\tau_0)].(\mu_{\text{in}}-I(x))d\Omega\right. \\ &\qquad \left. - \int_{R^c} \delta[(\mu_{\text{in}}-\mu_{\text{out}})(I(x)-\tau_0)].(\mu_{\text{in}}-I(x))d\Omega\right\}\nabla_C\mu_{\text{in}} \\ &+ \left\{\int_R \delta[(\mu_{\text{in}}-\mu_{\text{out}})(I(x)-\tau_0)].(I(x)-\mu_{\text{out}})d\Omega\right. \\ &\qquad \left. - \int_{R^c} \delta[(\mu_{\text{in}}-\mu_{\text{out}})(I(x)-\tau_0)].(I(x)-\mu_{\text{out}})d\Omega\right\}\nabla_C\mu_{\text{out}}\end{aligned} \tag{54}$$

Similarly to Section 3.3.1.1, one defines the curve Γ_0, parameterized with arclength s', as

$$\Gamma_0 = \{x \in \Omega \quad \text{s.t.} \quad I(x) = \tau_0 = \frac{\mu_{\text{in}}+\mu_{\text{out}}}{2}\}$$

Furthermore, one has

$$\nabla_x\big((\mu_{\text{in}}-\mu_{\text{out}})(I(x)-\tau_0)\big) = (\mu_{\text{in}}-\mu_{\text{out}})\nabla_x I.$$

Hence, using the generalized scaling-property of the Dirac delta function for multidimensional functionals, one gets

$$\begin{aligned}
\nabla_C E_{\text{image}} &= (1-2G).\mathbf{N} \\
&\quad + \frac{1}{|\mu_{\text{in}} - \mu_{\text{out}}|}\left\{\int_{R\cap\Gamma_0} \frac{\mu_{\text{in}} - I(x)}{\|\nabla_x I\|} ds' - \int_{R^c\cap\Gamma_0} \frac{\mu_{\text{in}} - I(x)}{\|\nabla_x I\|} ds'\right\} \nabla_C \mu_{\text{in}} \\
&\quad + \frac{1}{|\mu_{\text{in}} - \mu_{\text{out}}|}\left\{\int_{R\cap\Gamma_0} \frac{I(x) - \mu_{\text{out}}}{\|\nabla_x I\|} ds' - \int_{R^c\cap\Gamma_0} \frac{I(x) - \mu_{\text{out}}}{\|\nabla_x I\|} ds'\right\} \nabla_C \mu_{\text{out}} \\
&= (1-2G).\mathbf{N} \\
&\quad + \frac{1}{|\mu_{\text{in}} - \mu_{\text{out}}|}\left\{\int_{R\cap\Gamma_0} \frac{\mu_{\text{in}} - \mu_{\text{out}}}{2\|\nabla_x I\|} ds' - \int_{R^c\cap\Gamma_0} \frac{\mu_{\text{in}} - \mu_{\text{out}}}{2\|\nabla_x I\|} ds'\right\} \nabla_C \mu_{\text{in}} \\
&\quad + \frac{1}{|\mu_{\text{in}} - \mu_{\text{out}}|}\left\{\int_{R\cap\Gamma_0} \frac{\mu_{\text{in}} - \mu_{\text{out}}}{2\|\nabla_x I\|} ds' - \int_{R^c\cap\Gamma_0} \frac{\mu_{\text{in}} - \mu_{\text{out}}}{2\|\nabla_x I\|} ds'\right\} \nabla_C \mu_{\text{out}}
\end{aligned} \tag{55}$$

Defining the function $\text{sign}(\chi) = \begin{cases} 1, & \text{if } \chi > 0; \\ 0, & \text{if } \chi = 0; \\ -1, & \text{if } \chi < 0. \end{cases}$ and regrouping terms, one gets

$$\begin{aligned}
\nabla_C E_{\text{image}} &= (1-2G).\mathbf{N} + \text{sign}(\mu_{\text{in}} - \mu_{\text{out}})\left\{\int_{R\cap\Gamma_0} \frac{ds'}{\|\nabla_x I\|} - \int_{R^c\cap\Gamma_0} \frac{ds'}{\|\nabla_x I\|}\right\} \nabla_C \tau_0 \\
&= \underbrace{(1-2G).\mathbf{N}}_{\mathcal{V}_{\text{stat}}} + \underbrace{\beta.\nabla_C \tau_0}_{\mathcal{V}_{\text{geom}}}
\end{aligned} \tag{56}$$

where

$$\beta(\tau_0, C) = \text{sign}(\mu_{\text{in}} - \mu_{\text{out}})\left\{\int_{R\cap\Gamma_0} \frac{ds'}{\|\nabla_x I\|} - \int_{R^c\cap\Gamma_0} \frac{ds'}{\|\nabla_x I\|}\right\} \tag{57}$$

is a real number, and

$$\nabla_C \tau_0 = \frac{1}{2}(\nabla_C \mu_{\text{in}} + \nabla_C \mu_{\text{out}}) \tag{58}$$

with

$$\nabla_C \mu_{\text{in}} = \left(\frac{I - \mu_{\text{in}}}{A_{\text{in}}}\right)\mathbf{N} \qquad \text{and} \qquad \nabla_C \mu_{\text{out}} = \left(\frac{\mu_{\text{out}} - I}{A_{\text{out}}}\right)\mathbf{N} \tag{59}$$

The gradient in Equation (56) is the sum of two speed terms:

- The first term of ∇E_{image}, namely $\mathcal{V}_{\text{stat}} = (1-2G)\mathbf{N}$, simply points towards G: Image pixels (in the vicinity of C), which are more likely to belong to the Object are included inside C, whereas pixels more likely to belong to the Background are excluded. Hence, this term influences the contour based only on statistical considerations. Moreover, it is straightforward to realize that, for a given ϕ, the direction of $\mathcal{V}_{\text{stat}}$ and of the speed term in [12] (neglecting regularizing terms) are the same and these speed terms differ only by their magnitudes. The speed term $\mathcal{V}_{\text{stat}}$ can thus be expected to drive the contour in a similar fashion as the gradient proposed in [12], which also stems from statistical considerations of the intensity distributions only (This will be further highlighted in the experimental part).

- The second term of ∇E_{image}, namely $\mathcal{V}_{\text{geom}} = \beta \nabla \tau_0$, involves the gradient of the threshhold τ_0 (and, thus, also the gradients of the statistical moments μ_{in} and μ_{out}). Such a term is not present in the flows presented in the standard region-based technique [12] - see also Equation (49). The term $\mathcal{V}_{\text{geom}}$ increases or decreases the threshold τ_0, which in turn *results in changing* G, to decrease E_{image} as fast as possible. The direction and amplitude of the term $\mathcal{V}_{\text{geom}}$ is conditioned upon the coefficient β. Considering the expression of β in Equation (57), one can make the following remarks:

    1. The coefficient β compares the weighted lengths of the isolines Γ_0 inside $(R \cap \Gamma_0)$ and outside $(R^c \cap \Gamma_0)$ the curve. It is positive (respectively negative) when the weighted length of the inside-isoline $R \cap \Gamma_0$ is larger (respectively smaller) than the weighted length of the outside-isoline $R^c \cap \Gamma_0$. Also, the coefficient β is minimal when the isolines $R \cap \Gamma_0$ and $R^c \cap \Gamma_0$ coincide with C: Hence, more global information about the image, than traditional region-based techniques, is taken into account through β.

2. The coefficient β involves the gradients of the image $\nabla_x I$, similarly to edge-based techniques [10, 63]. However, the terms $\nabla_x I$ are summed over the length of Γ_0, making their influence less sensitive to noise (contrary to traditional edge-based approaches that compute edge maps locally), as is typical of region-based approaches. Also, the coefficient β receives less contribution from points of the isoline corresponding to strong edges (i.e.: for which $\nabla_x I$ high). This is in accordance with the intuition that the threshold should not be changed/moved too much if strong edges are found.

Hence, from the remarks above, the term $\mathcal{V}_{\text{geom}}$ evolves the curve C, based on geometric considerations of the the image and the contour: the threshold is adjusted until strong edges are found or until the edges of the thresholded image (the image isolines $R \cap \Gamma$ and $R^c \cap \Gamma$) coincide with C.

Thus, the speed term $\nabla_C E_{\text{image}}$ realizes a trade-off between purely a statistical force and a force that stems from geometric considerations.

3.4.1.2 Thresholding with two explicit thresholds

In [93, 106], a method is proposed to segment images composed of regions with distinct Gaussian densities, using the estimates

$$\begin{aligned} P_{\text{in}}(I, C) &= \mathcal{N}_{\mu_{\text{in}}, \Sigma_{\text{in}}}(I, C) = \frac{1}{\sqrt{2\pi\Sigma_{\text{in}}}} e^{-\frac{(I-\mu_{\text{in}})^2}{2\Sigma_{\text{in}}}} \quad \text{and} \\ P_{\text{out}}(I, C) &= \mathcal{N}_{\mu_{\text{out}}, \Sigma_{\text{out}}}(I, C) = \frac{1}{\sqrt{2\pi\Sigma_{\text{out}}}} e^{-\frac{(I-\mu_{\text{out}})^2}{2\Sigma_{\text{out}}}}. \end{aligned} \tag{60}$$

where the variances of the pixels located inside and outside the curve C are denoted by Σ_{in} and Σ_{out} respectively. The variances Σ_{in} and Σ_{out} are supposed to be distinct in this section (the case where $\Sigma_{\text{in}} = \Sigma_{\text{out}}$ is treated in the section above). The intensity averages μ_{in} and μ_{out} are computed as in section above, and the variances Σ_{in} and

Σ_{out} are computed at each step of the curve evolution as

$$\Sigma_{\text{in}}(C) = \frac{\int_R \left(I(x) - \mu_{\text{in}}\right)^2 d\Omega}{A_{\text{in}}} \text{ and } \Sigma_{\text{out}}(C) = \frac{\int_{R^c} \left(I - \mu_{\text{out}}\right)^2 d\Omega}{A_{\text{out}}}. \tag{61}$$

The flow in [93, 106] (Paragios-Deriche flow) is

$$\begin{aligned} \text{Paragios} - \text{Deriche flow} : \quad \frac{dC}{dt} &= \left(\log\left(P_{\text{in}}\right) - \log\left(P_{\text{out}}\right)\right).\mathbf{N} \\ &= \frac{1}{2}\log\left(\frac{\Sigma_{\text{out}}}{\Sigma_{\text{in}}}\right) + \frac{1}{2}\frac{((I-\mu_{\text{out}})^2}{\Sigma_{\text{out}}} - \frac{1}{2}\frac{((I-\mu_{\text{in}})^2}{\Sigma_{\text{in}}}).\mathbf{N} \\ &= \frac{\Sigma_{\text{in}} - \Sigma_{\text{out}}}{2\Sigma_{\text{in}}\Sigma_{\text{out}}}(I-\tau_1)(I-\tau_2).\mathbf{N} \end{aligned} \tag{62}$$

where the thresholds τ_1 and τ_2 can be computed as

$$\begin{aligned} \tau_1 &= \frac{\Sigma_{\text{out}}\mu_{\text{in}} - \Sigma_{\text{in}}.\mu_{\text{out}} - \alpha\sqrt{\Sigma_{\text{out}}\Sigma_{\text{in}}}}{\Sigma_{\text{out}} - \Sigma_{\text{in}}} \text{ and} \\ \tau_2 &= \frac{\Sigma_{\text{out}}\mu_{\text{in}} - \Sigma_{\text{in}}\mu_{\text{out}} + \alpha\sqrt{\Sigma_{\text{out}}\Sigma_{\text{in}}}}{\Sigma_{\text{out}} - \Sigma_{\text{in}}}, \text{ with} \\ \alpha &= \sqrt{(\mu_{\text{in}} - \mu_{\text{out}})^2 + (\Sigma_{\text{in}} - \Sigma_{\text{out}})\log(\frac{\Sigma_{\text{in}}}{\Sigma_{\text{out}}})}. \end{aligned} \tag{63}$$

Hence, according to the flow in Equation (62), the curve moves inward or outward depending on the sign of the function of the thresholds $\mathcal{T}_{PD}(x) = (\Sigma_{\text{in}} - \Sigma_{\text{out}})(I(x) - \tau_1)(I(x) - \tau_2)$.

Accordingly, we define

$$\begin{aligned} G(x) &= H\left(\log\left(P_{\text{in}}(I(x))\right) - \log\left(P_{\text{out}}(I(x))\right)\right) \\ &= H\left(\frac{\Sigma_{\text{in}} - \Sigma_{\text{out}}}{2\Sigma_{\text{in}}\Sigma_{\text{out}}}(I(x)-\tau_1)(I(x)-\tau_2)\right) \\ &= H\left[(\Sigma_{\text{in}} - \Sigma_{\text{out}})(I(x)-\tau_1)(I(x)-\tau_2)\right] = H[\mathcal{T}_{PD}(x)] \end{aligned} \tag{64}$$

where the scaling-property of the Heaviside function was used ($\frac{1}{2\Sigma_{\text{in}}\Sigma_{\text{out}}} > 0$). Contrary to Section 3.3.1.1, one thus has an explicit (closed-form) formula to compute the thresholds τ_1 and τ_2 at each step of the contour evolution.

The thresholds τ_1 and τ_2 are not in general optimal thresholds for the energy E_{image} given the contour C. Using the definition of τ_1 and τ_2 in Equation (63) above

(when minimizing E_{image} with respect to C) results in a constrained optimization of the form

$$\begin{cases} C^{**} = \arg\min_C \left(E_{\text{image}}(C, \tau)\right) \\ \tau(C) = (\tau_1, \tau_2) \end{cases} \tag{65}$$

The (constrained) minimization of E_{image} is carried via gradient descent. The Gradient of E_{image} with respect to the curve C can be computed, using Equation (30), as

$$\begin{aligned} \nabla_C E_{\text{image}} = (1 - 2G).\mathbf{N} \\ + \left\{ \int_R \delta[\mathcal{T}_{PD}(x)].(\Sigma_{\text{in}} - \Sigma_{\text{out}})(I(x) - \tau_2)d\Omega \right. \\ \left. - \int_{R^c} \delta[\mathcal{T}_{PD}(x)].(\Sigma_{\text{in}} - \Sigma_{\text{out}})(I(x) - \tau_2)d\Omega \right\} \nabla_C \tau_1 \\ + \left\{ \int_R \delta[\mathcal{T}_{PD}(x)].(\Sigma_{\text{in}} - \Sigma_{\text{out}})(I(x) - \tau_1)d\Omega \right. \\ \left. - \int_{R^c} \delta[\mathcal{T}_{PD}(x)].(\Sigma_{\text{in}} - \Sigma_{\text{out}})(I(x) - \tau_1)d\Omega \right\} \nabla_C \tau_2 \\ + \left\{ \int_{R^c} \delta[\mathcal{T}_{PD}(x)].(I(x) - \tau_1)(I(x) - \tau_2)d\Omega \right. \\ \left. - \int_R \delta[\mathcal{T}_{PD}(x)].(I(x) - \tau_1)(I(x) - \tau_2)d\Omega \right\} \nabla_C (\Sigma_{\text{in}} - \Sigma_{\text{out}}) \end{aligned} \tag{66}$$

Similarly to the sections above, one defines the curves Γ_1, parameterized with arclength s_1', and Γ_2, parameterized with arclength s_2' as

$$\Gamma_1 = \{x \in \Omega \quad \text{s.t.} \quad I(x) = \tau_1\} \quad \text{and} \quad \Gamma_2 = \{x \in \Omega \quad \text{s.t.} \quad I(x) = \tau_2\} \tag{67}$$

Furthermore, one has $\nabla_x \mathcal{T}_{PD} = (\Sigma_{\text{in}} - \Sigma_{\text{out}})(2I - \tau_1 - \tau_2)\nabla_x I$. Using the scaling property of the Dirac delta function generalized to multidimensional functionals, one

gets (the last term in Equation (66) above collapses)

$$\begin{aligned}\nabla_C E_{\text{image}} = \;& (1-2G).\mathbf{N} \\ & + \left\{ \int_{R\cap\Gamma_1} \frac{(\Sigma_{\text{in}}-\Sigma_{\text{out}})(\tau_1-\tau_2)}{\|(\Sigma_{\text{in}}-\Sigma_{\text{out}})(\tau_1-\tau_2)\nabla_x I\|} ds_1' \right. \\ & \qquad \left. - \int_{R^c\cap\Gamma_1} \frac{(\Sigma_{\text{in}}-\Sigma_{\text{out}})(\tau_1-\tau_2)}{\|(\Sigma_{\text{in}}-\Sigma_{\text{out}})(\tau_1-\tau_2)\nabla_x I\|} ds_1' \right\} \nabla_C \tau_1 \\ & + \left\{ \int_{R\cap\Gamma_2} \frac{(\Sigma_{\text{in}}-\Sigma_{\text{out}})(\tau_2-\tau_1)}{\|(\Sigma_{\text{in}}-\Sigma_{\text{out}})(\tau_2-\tau_1)\nabla_x I\|} ds_2' \right. \\ & \qquad \left. - \int_{R^c\cap\Gamma_2} \frac{(\Sigma_{\text{in}}+\Sigma_{\text{out}})(\tau_2-\tau_1)}{\|(\Sigma_{\text{in}}-\Sigma_{\text{out}})(\tau_2-\tau_1)\nabla_x I\|} ds_2' \right\} \nabla_C \tau_2 \end{aligned} \tag{68}$$

which, using the function sign, simplifies in

$$\begin{aligned}\nabla_C E_{\text{image}} \;=\; & (1-2G).\mathbf{N} \\ & + \; \text{sign}((\Sigma_{\text{in}}-\Sigma_{\text{out}})(\tau_1-\tau_2)) \left\{ \int_{R\cap\Gamma_1} \frac{ds_1'}{\|\nabla_x I\|} - \int_{R^c\cap\Gamma_1} \frac{ds_1'}{\|\nabla_x I\|} \right\} \nabla_C \tau_1 \\ & - \; \text{sign}((\Sigma_{\text{in}}-\Sigma_{\text{out}})(\tau_1-\tau_2)) \left\{ \int_{R\cap\Gamma_2} \frac{ds_2'}{\|\nabla_x I\|} - \int_{R^c\cap\Gamma_2} \frac{ds_2'}{\|\nabla_x I\|} \right\} \nabla_C \tau_2 \end{aligned} \tag{69}$$

Using the definition of τ_1 and τ_2 in Equation (63), it is straightforward to see that $\text{sign}((\Sigma_{\text{in}}-\Sigma_{\text{out}})(\tau_1-\tau_2)) = \text{sign}(2\alpha\sqrt{\Sigma_{\text{out}}\Sigma_{\text{in}}}) = +1$. Hence, one gets

$$\begin{aligned}\nabla_C E_{\text{image}} &= (1-2G).\mathbf{N} + \left\{ \int_{R\cap\Gamma_1} \frac{ds_1'}{\|\nabla_x I\|} - \int_{R^c\cap\Gamma_1} \frac{ds_1'}{\|\nabla_x I\|} \right\} \nabla_C \tau_1 \\ & \qquad - \left\{ \int_{R\cap\Gamma_2} \frac{ds_2'}{\|\nabla_x I\|} - \int_{R^c\cap\Gamma_2} \frac{ds_2'}{\|\nabla_x I\|} \right\} \nabla_C \tau_2 \\ &= \underbrace{(1-2G).\mathbf{N}}_{\mathcal{V}_{\text{stat}}} + \underbrace{\beta_1 \nabla_C \tau_1 - \beta_2 \nabla_C \tau_2}_{\mathcal{V}_{\text{geom}}} \end{aligned} \tag{70}$$

where

$$\begin{aligned} \beta_1(\tau_1, C) &= \left\{ \int_{R\cap\Gamma_1} \frac{ds_1'}{\|\nabla_x I\|} - \int_{R^c\cap\Gamma_1} \frac{ds_1'}{\|\nabla_x I\|} \right\} \qquad \text{and} \\ \beta_2(\tau_2, C) &= \left\{ \int_{R\cap\Gamma_2} \frac{ds_2'}{\|\nabla_x I\|} - \int_{R^c\cap\Gamma_2} \frac{ds_2'}{\|\nabla_x I\|} \right\} \end{aligned} \tag{71}$$

are real numbers.

We now detail the expressions of $\nabla_\phi I_1$ and $\nabla_\phi I_2$, which appear in Equation (70). Using classical rules of calculus, one gets

$$\begin{aligned}\nabla_C I_1 &= m_1^{\text{in}}.\nabla_C\mu_{\text{in}} + m_1^{\text{out}}.\nabla_C\mu_{\text{out}} + s_1^{\text{in}}.\nabla_C\Sigma_{\text{in}} + s_1^{\text{out}}.\nabla_C\Sigma_{\text{out}}.\\ \nabla_C I_2 &= m_2^{\text{in}}.\nabla_C\mu_{\text{in}} + m_2^{\text{out}}.\nabla_C\mu_{\text{out}} + s_2^{\text{in}}.\nabla_C\Sigma_{\text{in}} + s_2^{\text{out}}.\nabla_C\Sigma_{\text{out}}.\end{aligned} \tag{72}$$

where the parameters m_1^{in}, m_1^{out}, s_1^{in} and s_1^{out} are

$$\begin{aligned} m_1^{\text{in}} &= \frac{1}{\Sigma_{\text{out}} - \Sigma_{\text{in}}}\left(\Sigma_{\text{out}} - \sqrt{\Sigma_{\text{in}}\Sigma_{\text{out}}}\left(\frac{\mu_{\text{in}} - \mu_{\text{out}}}{\alpha}\right)\right)\\ m_1^{\text{out}} &= \frac{1}{\Sigma_{\text{out}} - \Sigma_{\text{in}}}\left(\sqrt{\Sigma_{\text{in}}\Sigma_{\text{out}}}\left(\frac{\mu_{\text{in}} - \mu_{\text{out}}}{\alpha}\right) - \Sigma_{\text{in}}\right)\end{aligned}$$

$$\begin{aligned} s_1^{\text{in}} &= \frac{1}{\Sigma_{\text{out}} - \Sigma_{\text{in}}}\left\{\frac{\Sigma_{\text{out}}\mu_{\text{in}} - \Sigma_{\text{out}}\mu_{\text{out}} - \alpha\sqrt{\Sigma_{\text{in}}\Sigma_{\text{out}}}}{\Sigma_{\text{out}} - \Sigma_{\text{in}}}\right.\\ &\qquad \left. - \left(\frac{\sqrt{\Sigma_{\text{in}}\Sigma_{\text{out}}}}{2\alpha}\log\left(\frac{\Sigma_{\text{in}}}{\Sigma_{\text{out}}}\right) + \left(\frac{\Sigma_{\text{in}} - \Sigma_{\text{out}}}{2\alpha} + \frac{\alpha}{2}\right)\sqrt{\frac{\Sigma_{\text{out}}}{\Sigma_{\text{in}}}}\right)\right\}\\ s_1^{\text{out}} &= \frac{1}{\Sigma_{\text{out}} - \Sigma_{\text{in}}}\left\{\frac{\Sigma_{\text{in}}\mu_{\text{out}} - \Sigma_{\text{in}}\mu_{\text{in}} + \alpha\sqrt{\Sigma_{\text{in}}\Sigma_{\text{out}}}}{\Sigma_{\text{out}} - \Sigma_{\text{in}}}\right.\\ &\qquad \left. - \left(\frac{\sqrt{\Sigma_{\text{in}}\Sigma_{\text{out}}}}{2\alpha}\log\left(\frac{\Sigma_{\text{out}}}{\Sigma_{\text{in}}}\right) + \left(\frac{\Sigma_{\text{out}} - \Sigma_{\text{in}}}{2\alpha} + \frac{\alpha}{2}\right)\sqrt{\frac{\Sigma_{\text{in}}}{\Sigma_{\text{out}}}}\right)\right\}\end{aligned} \tag{73}$$

The expressions for the parameters m_2^{in}, m_2^{out}, s_2^{in} and s_2^{out} are identical to the expressions of m_1^{in}, m_1^{out}, s_1^{in} and s_1^{out}, replacing α by $-\alpha$.

The expressions of $\nabla_C\Sigma_{\text{in}}$ and $\nabla_C\Sigma_{\text{out}}$ are given by

$$\nabla_C\Sigma_{\text{in}} = \left(\frac{(I - \mu_{\text{in}})^2 - \Sigma_{\text{in}}}{A_{\text{in}}}\right)\mathbf{N} \quad \text{and} \qquad \nabla_C\Sigma_{\text{out}} = \left(\frac{\Sigma_{\text{out}} - (I - \mu_{\text{out}})^2}{A_{\text{out}}}\right)\mathbf{N}. \tag{74}$$

The gradient in Equation (56) is the sum of two types of speed terms: a purely statistical term ($(1-2G).\mathbf{N}$) and two geometric ones ($\beta_1\nabla_C\tau_1$ and $\beta_2\nabla_C\tau_2$) that evolve the thresholds τ_1 and τ_2 to find strong edges or to make the edges of the thresholded image (curves Γ_1 and Γ_2) coincide maximally with C.

3.4.1.3 Implementation/Fuzzy Thresholding

In this part, we show how one can perform optimization with respect to the level-set function ϕ. This is in contrast with the computations that were performed in the sections above where gradients, for instance, were directly computed with respect to the curve C. Gradient computations with respect to ϕ were popularized in [12], where the Heaviside and Dirac delta functionals are defined in the distributional sense through (smooth) sigmoid functions, for instance. Flows based on gradients computed with respect to C or ϕ are not strictly equivalent in general. Rather, flows obtained from gradients computed with respect to ϕ can be considered as approximations of flows that result from gradients computed with respect to C. However, the resulting contour evolutions are usually similar notably when using a narrow-band implementation [116]. The advantage of using flows computed with respect to ϕ in association with mollified versions of the Heaviside and Dirac delta functions is that computations and implementation are usually facilitated.

A smooth version of H, denoted $H_\epsilon : \mathbb{R} \mapsto [0,1]$, can be computed as follows for a chosen parameter ϵ:

$$H_\epsilon(\chi) = \begin{cases} = 1 & \text{if } \chi > \epsilon \ ; \\ = 0 & \text{if } \chi < -\epsilon \ ; \\ = \frac{1}{2}\{1 + \frac{\chi}{\epsilon} + \frac{1}{\pi}\sin\left(\frac{\pi\chi}{\epsilon}\right)\} & \text{otherwise.} \end{cases} \tag{75}$$

The derivative of H_ϵ will be denoted by δ_ϵ and computed as

$$\delta_\epsilon(\chi) = \begin{cases} = \frac{1}{2\epsilon}\{1 + \cos\left(\frac{\pi\chi}{\epsilon}\right)\} & \text{if } |\chi| < \epsilon \ ; \\ = 0 & \text{otherwise.} \end{cases} \tag{76}$$

The smooth version H_ϵ of H presented above can be used as a labeling function to describe shapes. Indeed, and in contrast with the sections above where shapes are

represented via strict binary maps, a shape $\mathcal{S} \subset \Omega$ can also be characterized by a *smooth* labeling function $\mathrm{S}_{\mathcal{S}} : \Omega \mapsto [0,1]$ such as

$$\mathrm{S}_{\mathcal{S}}(x) \begin{cases} \geq \frac{1}{2} & \text{if } x \in \mathcal{S};; \\ < \frac{1}{2} & \text{if } x \in \Omega \backslash \mathcal{S}. \end{cases} \tag{77}$$

For instance, the shape defined by the interior of the contour C can now be characterized by $H_{\epsilon_1}\phi$, for a chosen (small) value of the parameter $\epsilon_1 \in \mathbb{R}$ (N.B.: In the remainder, we will usually omit ϵ_1 in the expressions of $H_{\epsilon_1}(\phi)$ and $\delta_{\epsilon_1}(\phi)$ and denote $H\phi$ and $\delta\phi$ to simplify the notation).

The energy E_{image}, as defined in Equation (29), can be written using mollified Heaviside functions. In particular the function G, defined in Equation (46), becomes $G = H_{\epsilon_2}\big[\log\big(P_{\text{in}}\big) - \log\big(P_{\text{out}}\big)\big]$, for a certain (small) value of the parameter $\epsilon_2 \in \mathbb{R}$. The use of mollified approximations of the Heaviside function, which attribute a value in the range $[0,1]$ to the each location of the maps $H\phi$ and G, gives rise to an alternative interpretation of the energy E_{image} as a measure of the discrepancy between two confidence maps:

- The (smooth) map $H\phi$ can be seen as a measure of confidence that pixels belong to the Object or the Background, based on spatial considerations. For instance, the further away pixels are from the zero level-set while being inside the curve (respectively, outside), the closer to 1 (respectively, 0) is the value that they are attributed through $H\phi$. A value closer to 1 (respectively, 0) indicates that the pixel is more likely to belong to the Object (respectively, to the Background). Thus, in the function $H\phi$, the uncertainty is spatial only and the most uncertain pixels are around the zero level-set (which are attributed a confidence measure around $\frac{1}{2}$).

- The (smooth) map G can be seen as a measure of confidence that pixels belong to the Object or the Background, based on statistical considerations. For instance,

for a pixel x the larger is $P_{\text{in}}(I(x))$ is as compared to $P_{\text{out}}(I(x))$, the closer to 1 is the value $G(x)$. Thus, in the function G, the uncertainty is statistical only and the most uncertain pixels x are such that $P_{\text{in}}(I(x)) \simeq P_{\text{out}}(I(x))$ (which are attributed a confidence measure around $\frac{1}{2}$). Thus, the function G behave similarly as a **fuzzy-thresholding** function.

Hence, when smooth approximation of H are used, the energy E_{image} can be seen as realizing a trade-off between two types of confidence measures: spatial ones (through $H\phi$) and statistical ones (through G). As a practical result, the two parameters ϵ_1 and ϵ_2 can be adjusted to improve segmentation performance, in similar ways that fuzzy thresholding approaches can improve (hard) thresholding techniques (see e.g. [59] and the references therein)

The different flows in Equations (56) and (70) can be expressed with respect to ϕ (instead of C), and smooth version of $H\phi$ and G. For instance the flow in Equation (56) becomes

$$\nabla_\phi E_{\text{image}} = 2.\delta\phi\big(H\phi - G\big) + 2\beta.\nabla_\phi\tau, \tag{78}$$

where the expressions of $\beta \in \mathbf{R}$ and $\nabla_\phi\tau : \Omega \mapsto \mathbf{R}$ are

$$\begin{aligned} \beta &= \text{sign}(\mu_{\text{in}} - \mu_{\text{out}}) \int_\Omega \delta_{\epsilon_2}\big(I(x) - \tau\big)[H\phi(x) - G(x)]dx \\ \nabla_\phi\tau &= \frac{1}{2}\big(\frac{\mu_{\text{out}} - I}{A_{\text{out}}} + \frac{I - \mu_{\text{in}}}{A_{\text{in}}}\big)\delta\phi \end{aligned} \tag{79}$$

We refer to [29], for the expression of Equation (70), in terms of ϕ.

N.B.: Using a mollified version δ_{ϵ_2} of the Dirac delta function as defined in Equation (76) for the derivative of G, offers an alternative way[4] to make the flow well defined since the condition $\nabla_x I \neq 0$ is no longer necessary.

[4] i.e., alternative to the approach presented in Section 3.3.1.1 using quantized thresholds in $\mathcal{Z}_{\frac{1}{2}}$

3.4.1.4 Thresholding with implicit thresholds

We now present generalizations of the method, in which the thresholds are not explicitly expressed as in Sections 3.4.1.1 and 3.4.1.2. Although the interpretation of the flow is harder than the case when thresholds are computed explicitly, using implicit thresholds allows one to extend the methodology to larger classes of images such as color and texture images or images with multi-modal intensity distribution that cannot be modeled satisfyingly with Gaussian densities as presented above. We first present a generalization of the Gaussian case treated above. Then we present how the flow can be further generalized by using a non-parametric density estimation approach for the densities inside and outside the curve, for instance.

Gaussian Case: Let us use normal distributions to model the conditional densities P_{in} and P_{out}. We focus in this section on highlighting the differences that occur when dealing with color images as compared to greyscale images. Other vector valued images (texture, diffusion tensor...) can be accommodated similarly. In the case of vector-valued images, the parametric densities can be computed as

$$\begin{aligned} P_{\text{in}}(I,\phi) &= \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_{\text{in}}|^{\frac{1}{2}}} e^{-\frac{1}{2}(I-\mu_{\text{in}})^T\Sigma_{\text{in}}^{-1}(I-\mu_{\text{in}})} \quad \text{and} \\ P_{\text{out}}(I,\phi) &= \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_{\text{out}}|^{\frac{1}{2}}} e^{-\frac{1}{2}(I-\mu_{\text{out}})^T\Sigma_{\text{out}}^{-1}(I-\mu_{\text{out}})}. \end{aligned} \tag{80}$$

where the intensity averages μ_{in} and μ_{out} of the pixels located inside and outside the curve C can be computed as functionals of ϕ as

$$\mu_{\text{in}}(\phi) = \frac{\int_\Omega I(x)H\phi(x)\,dx}{A_{\text{in}}} \text{ and } \mu_{\text{out}}(\phi) = \frac{\int_\Omega I(x)(1-H\phi(x))\,dx}{A_{\text{out}}}, \tag{81}$$

with the areas inside and outside the curve, expressed as functional of ϕ are given by

$$A_{\text{in}} = \int_\Omega H\phi(x)dx \quad \text{and} \quad A_{\text{out}} = \int_\Omega \big(1-H\phi(x)\big)dx$$

Similarly the variances inside (Σ_{in}) and outside (Σ_{out}) the curve are computed with

respect to ϕ as

$$\begin{aligned}\Sigma_{\text{in}}(\phi) &= \frac{\int \big(I(x)-\mu_{\text{in}}\big)\big(I(x)-\mu_{\text{in}}\big)^T H\phi(x)\ dx}{A_{\text{in}}} \text{ and} \\ \Sigma_{\text{out}}(\phi) &= \frac{\int \big(I-\mu_{\text{out}}\big)\big(I-\mu_{\text{out}}\big)^T \big(1-H\phi\big)\ dx}{A_{\text{out}}}.\end{aligned} \tag{82}$$

For grayscale images $n = 1$, $\mu_{\text{in/out}}(\phi) \in \mathbb{R}$ and $\Sigma_{\text{in/out}}(\phi) \in \mathbb{R}$. For color images $n = 3$, $\mu_{\text{in/out}}(\phi) \in \mathbb{R}^3$ and $\Sigma_{\text{in/out}}(\phi) \in \mathbb{R}^{3\times 3}$.

Using the definition of P_{in} and P_{out} above and Equation (46), the notion of thresholding, which results in building G, is straightforwardly extended to color images. The gradient $\nabla_\phi E_{\text{image}}$ can be computed using calculus of variations taking into account the variation of **both** $H\phi$ and G with respect to ϕ. We detail derivation in the Appendix B.1, Section B.3, and simply state the result for $\nabla_\phi E_{\text{image}}$:

$$\nabla_\phi E_{\text{image}} = \underbrace{\delta\phi.(H\phi - G)}_{\mathcal{V}_{\text{stat}}} + \underbrace{\beta_\mu^{\text{out}}.\nabla_\phi\mu_{\text{out}} + \beta_\Sigma^{\text{out}}.\nabla_\phi\Sigma_{\text{out}} - \beta_\mu^{\text{in}}.\nabla_\phi\mu_{\text{in}} - \beta_\Sigma^{\text{in}}.\nabla_\phi\Sigma_{\text{in}}}_{\mathcal{V}_{\text{geom}}} \tag{83}$$

where the expressions of the (constant) coefficients β_μ^{in} and β_Σ^{in} are given by

$$\begin{aligned}\beta_\mu^{\text{in}} &= \int_\Omega \gamma(u)\Sigma_{\text{in}}^{-1}(I(u)-\mu_{\text{in}})du \text{ and} \\ \beta_\Sigma^{\text{in}} &= \frac{1}{2}\int_\Omega \gamma(u).(\Sigma_{\text{in}}^{-1}(I(u)-\mu_{\text{in}})(I(u)-\mu_{\text{in}})^T\Sigma_{\text{in}}^{-1} - \Sigma_{\text{in}}^{-1})du\end{aligned} \tag{84}$$

with

$$\gamma(x) = (H\phi(x) - G(x)).\delta\left(\log\big(\frac{P_{\text{in}}(I(x))}{P_{\text{out}}(I(x))}\big)\right)$$

For gray-scale images $\beta_\mu^{\text{in}} \in \mathbb{R}$ and $\beta_\Sigma^{\text{in}} \in \mathbb{R}$; for color images $\beta_\mu^{\text{in}} \in \mathbb{R}^3$ and $\beta_\Sigma^{\text{in}} \in \mathbb{R}^9$ (row vectors). The expressions of β_μ^{out} and $\beta_\Sigma^{\text{out}}$ can be computed by simply replacing μ_{in} with μ_{out} and Σ_{in} with Σ_{out} in Equation (84). The expressions of the gradients of

the statistical moments are

$$\begin{aligned}
&\nabla_\phi \mu_{\text{in}} = \delta\phi.\left(\frac{I-\mu_{\text{in}}}{A_{\text{in}}}\right) \quad , \quad \nabla_\phi \mu_{\text{out}} = \delta\phi.\left(\frac{\mu_{\text{out}}-I}{A_{\text{out}}}\right), \\
&\nabla_\phi \Sigma_{\text{in}} = \delta\phi.\underbrace{\left(\frac{(I-\mu_{\text{in}})(I-\mu_{\text{in}})^T-\Sigma_{\text{in}}}{A_{\text{in}}}\right)}_{S_{\text{in}}} \quad \text{and} \\
&\nabla_\phi \Sigma_{\text{out}} = \delta\phi.\underbrace{\left(\frac{\Sigma_{\text{out}}-(I-\mu_{\text{out}})(I-\mu_{\text{out}})^T}{A_{\text{out}}}\right)}_{S_{\text{out}}}.
\end{aligned} \tag{85}$$

In the case of color images, $S_{\text{in}}(x)$ and $S_{\text{out}}(x)$ in the equation above are elements of $\mathbb{R}^9$ (column vectors).

Once again, the gradient in Equation (83) is the sum of two speed terms: a purely statistical term $\mathcal{V}_{\text{stat}}$ and a geometric one that involves the gradient of the statistical moments $\mathcal{V}_{\text{geom}}$. The term $\mathcal{V}_{\text{geom}}$ influences the contour evolution based on the number *ambiguous pixels*[5], i.e., pixels for which intensities I^* are such that $P_{\text{in}}(I^*) \simeq P_{\text{out}}(I^*)$ and for which classification is the most uncertain (Such ambiguous pixels can be visualized on the third row of Figure 11).

Regions of different intensity histograms: We should note that the proposed framework is general in the sense that several other families of distributions in addition to the Gaussian case could also be used. See Appendix B.4 for the use of non-parametric Parzen windows estimators [94] to model the pixel densities inside and outside the curve, as was also done in [64, 100] for instance.

Further, one can use average intensities only, as in [12], by replacing Σ_{in} and Σ_{out} by identity matrices.

[5]The functional $\delta_{\epsilon_2}\left(\log\left(\frac{P_{\text{in}}(I(x))}{P_{\text{out}}(I(x))}\right)\right)$ is indeed close to zero almost everywhere except at the ambiguous pixels.

3.4.2 Experiments for the Joint Geometric and Statistical Case

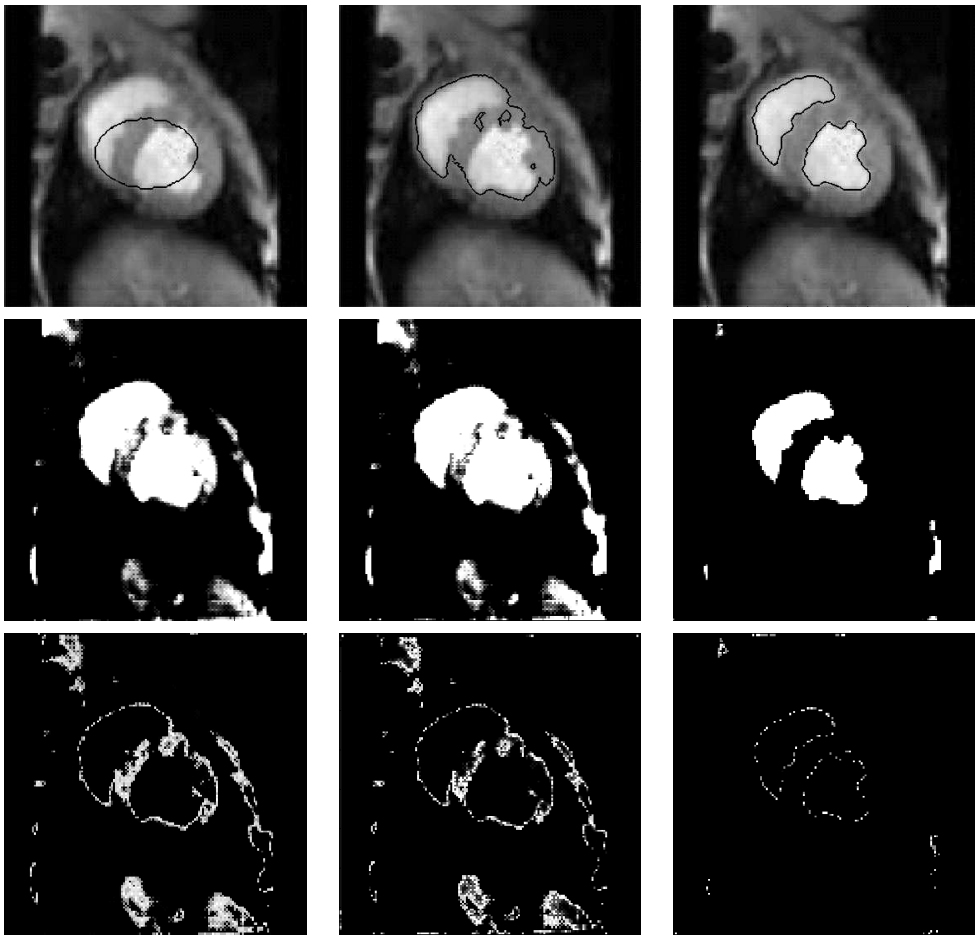
We now present experimental results for our approach combining geometric and statistical information. In our implementation of Equation (78), which approximates Equation (56), we used the heuristic $\epsilon_2 = 5 \times \epsilon_1$ for greyscale images $I : \Omega \mapsto \{0..255\}$. In our implementation of Equation (78), which approximates Equation (70) for greyscale image, we used the heuristic $\epsilon_2 = 3 \times \epsilon_1$ for greyscale images and $\epsilon_2 = \frac{\epsilon_1}{3}$ for color images. The curve evolution was performed by evolving C according to the flow:

$$\frac{d\phi}{dt} = -\nabla_\phi E_{\text{image}} + \lambda.\delta\phi.\text{div}\left(\frac{\nabla\phi}{\|\nabla\phi\|}\right) \tag{86}$$

where the second term in the right-hand side is a regularizing term penalizing high curvatures (the parameter λ is chosen subjectively in the range [0,2]).

We mostly compare our results with the results obtained using the method in [12] and [106]. This is motivated by the fact that these methods are arguably among the standard and most used methods in the field, as well as the fact that we made the same hypotheses concerning the distributions of pixel intensities (Gaussian distributions) and used the same criterion to distinguish between densities (log-likelihood). This allows us to compare approaches in a meaningful manner and to highlight the particularities of our flow.

We begin with the example of the heart image used above to visually illustrate some of the key ideas alluded to, in the analysis of the flow made in Section 3.4.1.1. In Figure 11, the flow of Section 3.4.1.1 is used. A convincing segmentation of the left ventricle is obtained with our method, for both the the initialization in the figure (for which the purely geometric approach over-segmented the ventricle) and the one in Figure 7 (for which the purely geometric approach successfully segmented the ventricle). This shows that the joint statistical and geometric flow preserves the enviable properties of the purely geometric flow as seen in Section 3.3.2 and is less



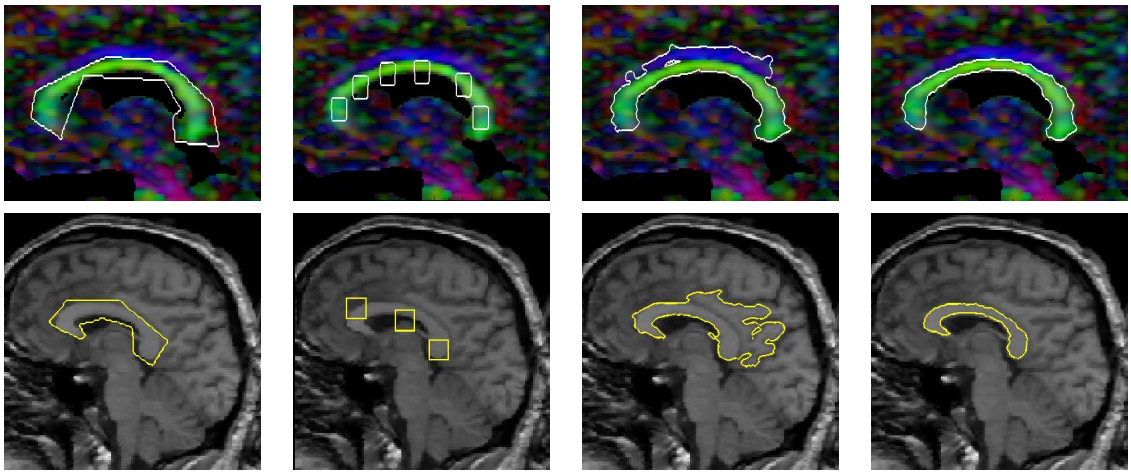
(a) (b) (c)

Figure 11: Segmentation of an MRI image of the heart. *Top Row*: (a)-Initialization; (b)-Final segmentation with the method in [106] or $\mathcal{V}_{\text{stat}}$ only; (c)-Final segmentation with the proposed method. *Middle Row*: Corresponding thresholded image ($G(x)$) with the threshold τ_0. *Bottom Row*: Corresponding isoline Γ_0 see Section 3.4.1.1.

sensitive to initialization, at least for this example. Using the methods proposed in [12, 106], the ventricle is not properly segmented. The same result as the ones obtained with [12, 106] was reached using the first term of $\nabla_\phi E_{\text{image}}$, i.e., $\mathcal{V}_{\text{stat}}$ only; this is done simply taking $\mathcal{V}_{\text{geom}} = 0$. This fact elucidates the influence of $\mathcal{V}_{\text{geom}}$ in constraining the curve evolution towards a successful segmentation. In particular, we note that, initially, the isoline Γ_0 as defined in Section 3.4.1.1 (third row in the figure) is in majority outside the curve C: the weighed length of the curve $R^c \cap \Gamma_0$ is larger than the length of the curve $R \cap \Gamma_0$. Comparing the final isoline Γ_0, whether our full

flow or $\mathcal{V}_{\text{stat}}$ only was used, one can note that the curve Γ_0 was changed drastically under the influence of $\mathcal{V}_{\text{geom}}$ (This also results in a controlled modification of G, c.f. second line in the figure). As can be noticed on the figure, the term $\mathcal{V}_{\text{geom}}$ indeed gathers the isolines $R^c \cap \Gamma_0$ and $R \cap \Gamma_0$ on C, as is expected from the analysis of the flow in Section 3.4.1.1. The final contour coincides maximally with Γ_0 that itself coincides with a region of high gradient in the image (strong edge), when our full flow is used. In addition, we also notice that an accurate segmentation is obtained using only intensity averages with our method, whereas a poor segmentation result is obtained with the method in [106] that uses intensity average *and* standard deviation. This highlights the importance of geometric information obtained from $\mathcal{V}_{\text{geom}}$ over (more complex) statistical models. We note that a successful segmentation of this image was also obtained in [135], where global constraints were imposed on the (statistical) flow presented in [123]. The geometric information exploited by our method can be seen as a way to also constrain the statistical information extracted from the image.

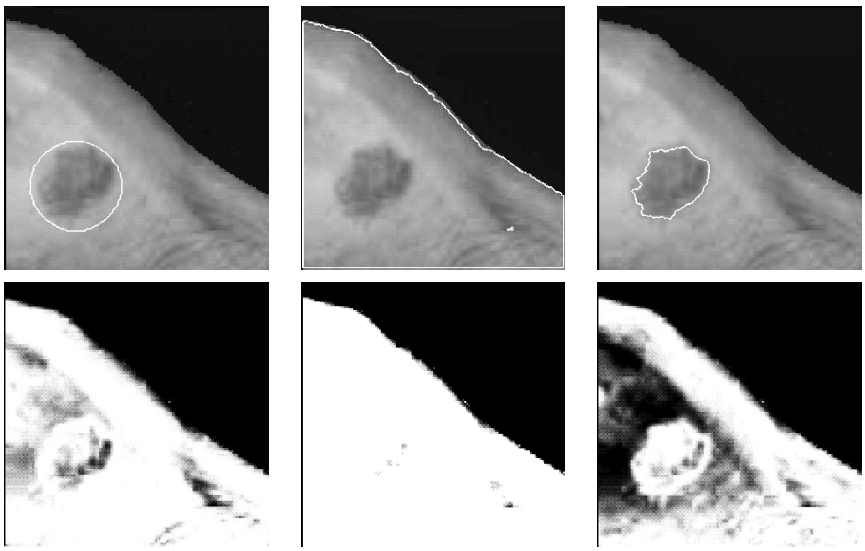
Segmenting medical images is many times a challenging task, since structures of interest are often poorly contrasted with respect to other neighboring structures. The proposed method was found to perform in a convincing manner on such images. In Figure 12, a color-coded fractional anisotropy and a gray-scale MRI images of the brain are presented. Reasonable segmentations of the corpus callosum (e.g., green structure in the color image) are obtained for two different initializations (close and far form the structure), with our technique (the flow in Section 3.4.1.4 was used on the color image and the flow in Section 3.4.1.1 was used on the greyscale image). Using the flows proposed in [106, 12] (or $\mathcal{V}_{\text{stat}}$ only), the contour leaks and does not capture the structure correctly (even when the initialization is close to the truth). In Figure 13, the hand of a patient with a Kaposi Sarcoma (KS) is segmented. The contour is initialized in the vicinity of the KS, and an acceptable segmentation of the pathology is obtained with our method (using flow in Section 3.4.1.2). Using the flow proposed



(a) (b) (c) (d)

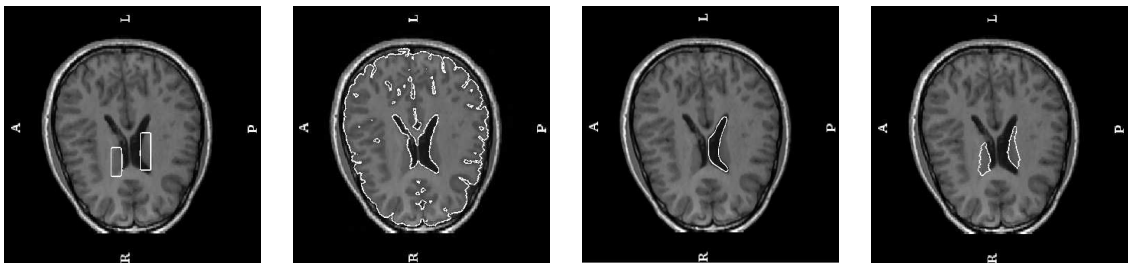
Figure 12: Comparative segmentation of the corpus callosum in a color-coded Diffusion Tensor MRI image of the brain, for diverse initializations (a)-Initialization ♯1 (close to the object); (b)-Initialization ♯2 (far from the object); (c)-Final segmentation with the methods in [106, 12] or $\mathcal{V}_{\text{stat}}$ only; (d)-Final segmentation with the proposed method.

in [106], for the same initialization, the contour ends up capturing the whole visible part of the hand. A similar result was obtained using the first term in ∇E_{image} only. In Figure 14, a transverse MRI-image of the brain is presented. The caudate nuclei are notoriously difficult structures to segment due to their poor contrast with neighboring structures. Using the method in [106] (or $\mathcal{V}_{\text{stat}}$ only), the whole white and gray matter of the brain is segmented, even though the contour was initialized in the vicinity of the caudate nuclei. Using another purely statistical technique that uses *global intensity histograms* (e.g., [78]) the contour leaks in the neighboring darker structure and does not segment the caudate. An acceptable segmentation of the left and right caudate is obtained with our method (using flow in Section 3.4.1.2). Hence, even using a more involved statistical technique (global pixel intensity histograms) than the rather simple Gaussian model in our approach, the structure is not satisfyingly segmented. This further illustrates the importance of the geometric information obtained from $\mathcal{V}_{\text{geom}}$. Figure 15 presents the results obtained for diverse MRI images (brain: corpus callosum; heart: left ventricle). On these examples again, the contour leaks into



(a) (b) (c)

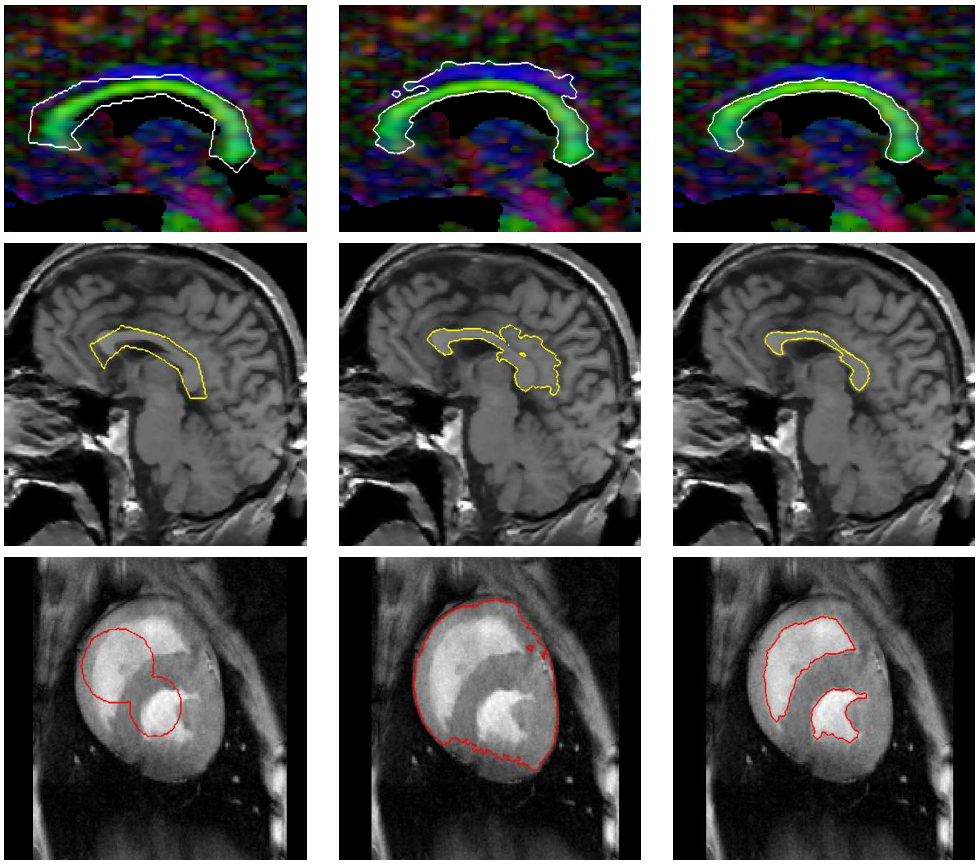
Figure 13: Segmentation of the hand of a patient with a Kaposi Sarcoma. *Top Row*: (a)-Initialization; (b)-Final segmentation with the method proposed in [106] or with the first term of ∇E_{image} only; (c)-Final segmentation with the proposed method.*Bottom Row*: Corresponding thresholded images.



(a) (b) (c) (d)

Figure 14: Comparative segmentation of the caudate nuclei. (a)-Initialization (close to the structure); (b)-Final segmentation with the methods in [106, 12] or $\mathcal{V}_{\text{stat}}$ only; (c)-Final segmentation with the method in [78] that uses global intensity histograms (a more involved model of pixels statistics than the Gaussian model used in our technique); (d)-Final successful segmentation with the proposed method.

neighboring structures, when the flow proposed in [12] and [106] is used. A common approach to improve these results would be to learn the shapes of the regions of interest a priori and to constrain the contour evolution with shape priors [70, 26]. In each case and *without the use of shape prior*, the regions of interest are successfully segmented using our approach.



(a) (b) (c)

Figure 15: Segmentations medical images. (a)-Initialization; (b)-Final segmentation with the method in [106] or $\mathcal{V}_{\text{stat}}$ only.; (c)-Final segmentation with the proposed method.

We now present results obtained for artificial and natural images. Figure 16 presents segmentation results (using the flow in Section 3.4.1.1) on the synthetic image used in Section 3.3.2. The word "Yellow" is successfully segmented. It is interesting to note how the thresholded images G in Figure 7 (purely geometric case) and in Figure 16 (joint statistical and geometric case) differ. In Figure 16, the contour evolves

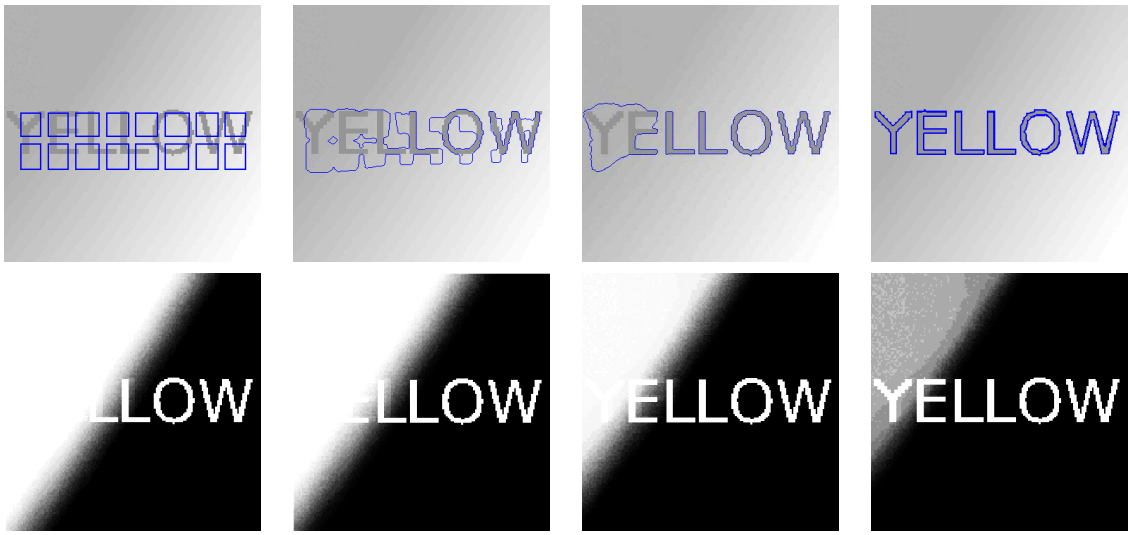


Figure 16: Synthetic image, word with linear background. *First line:* Successive steps of the contour evolution using the flow in Section 3.4.1.1. *Second line:* Corresponding thresholded image.

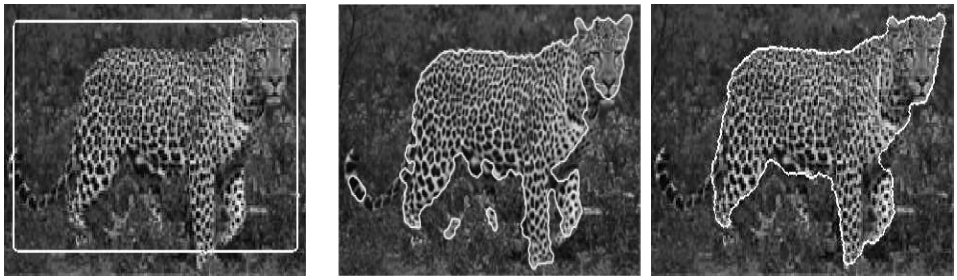
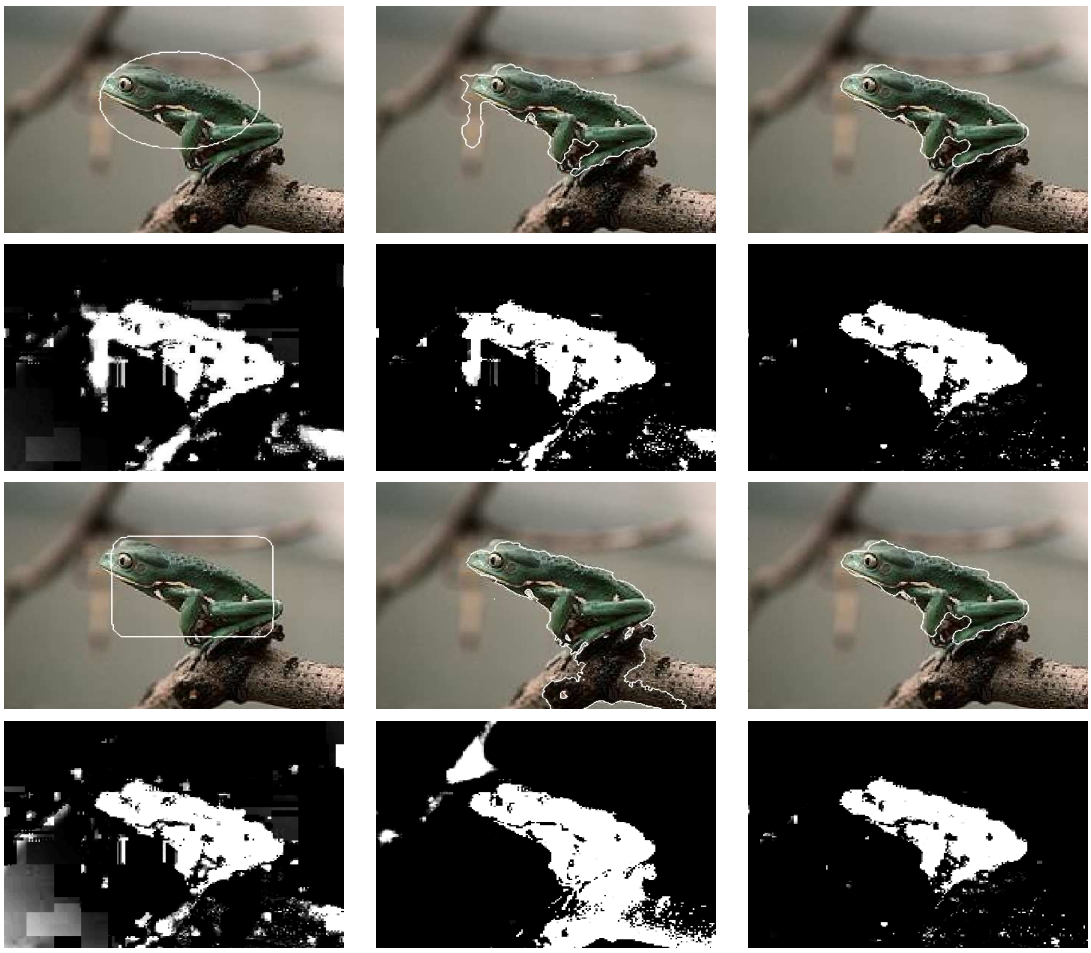


Figure 17: Segmentation results of a natural textured image. *Left*: Initialization, *Middle*: segmentation obtained with the method in [64] that uses mutual information and global intensity histograms (reproduced from [64]), *Right*: Segmentation result with the proposed method, using a simple Gaussian model.

to eliminate the extra part of G until it almost corresponds to the thresholded image of Figure 7. Again, the enviable properties of the purely geometric flow appear to be preserved by the joint statistical and geometric flow. In Figure 17, a textured natural image is segmented. The segmentation of the leopard obtained with our method is convincing compared to the segmentation obtained in [64], where an involved statistical model was used (the mutual information between general distributions of pixel intensities). In Figure 18, a frog is segmented for two different initializations. In



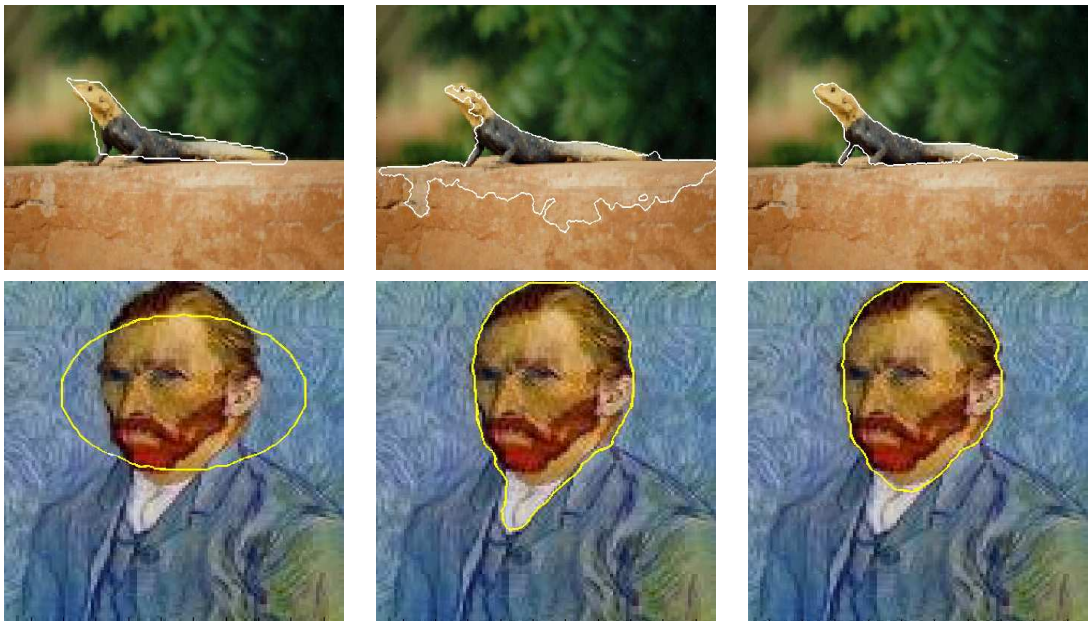
(a) (b) (c)

Figure 18: Segmentation of a green frog for two initializations. 1st *Row*: (a)-First Initialization; (b)-Final segmentation with the method in [106] or $\mathcal{V}_{\text{stat}}$ only; (c)-Final segmentation with the proposed method.2nd *Row*: Corresponding image shape models, $G(x)$. 3rd *Row*: (a)- Second Initialization; (b)-Final segmentation with the proposed method; (c)-Final segmentation with the method in [106] or $\mathcal{V}_{\text{stat}}$ only. 4th *Row*: Corresponding image shape models, $G(x)$.

both cases, a successful segmentation is obtained using our technique, whereas using the method in [106] the final contour ends up capturing significant parts of the background (branches or trunk). Using $\mathcal{V}_{\text{stat}}$ only, the same results as [106] are obtained. Using the flow in [106] (or $\mathcal{V}_{\text{stat}}$ only), for the first initialization the underlying thresholded image is little modified from initialization to convergence (second row in the figure), whereas for the second initialization the underlying thresholded image is allowed to drift (fourth row in the figure). In both cases, using our complete flow, the thresholded image is controlled and modified to lead to the same final result. This highlights again the influence of $\mathcal{V}_{\text{geom}}$ in constraining the curve evolution towards a successful segmentation. Furthermore, the same satisfying result is obtained for two initializations with our method, which illustrates its robustness to initialization.

In Figure 19, a lizard and a Van Gogh painting are segmented. The lizard is correctly segmented with our method. Using the flow in [106], the head of the lizard is not correctly segmented and the contour leaks into the wall, despite the fact that the contour was initially positioned very close to the animal. In the case of the painting, both methods lead to satisfying segmentations of the head of the subject. However, the contour was initialized as to exclude the person's white collar. Our method preserves this, whereas the method proposed in [106] also captures the white collar. These two examples show a valuable feature of our flow: information present in the initialization is more accurately preserved.

In Figure 20, the robustness of the method to noise is tested on two artificial and two natural images. The two artificial images were taken from [106] and are composed of two regions purely built from Gaussian noise: for the first artificial image, auto-correlation coefficients (between color components) are different, while for the second artificial image only cross-correlation coefficients are different. Contrary to more classical approaches such as the one presented in [11], our approach is able to deal with these differences. For the two natural images, Gaussian noise of mean 0 and



(a) (b) (c)

Figure 19: Segmentation of a lizard and a painting. (a)-Initialization; (b)-Final segmentation with the method in [106] or $\mathcal{V}_{\text{stat}}$ only; (c)-Final segmentation with the proposed method.

variance 5% and 10% were respectively added.

In Figure 21, various natural images with texture and noise are successfully segmented with our technique, highlighting the breadth of application of the method.

Hence, in view of the experiments performed, our method appears to lead to meaningful segmentations notably in the case of medical images. When only the first term of ∇E_{image} is used, similar results as [106] or [12] were obtained. This highlights the influence of the second term in ∇E_{image}. This term evolves the contour in ways that modify G to match $H\phi$ as fast as possible (by changing the statistics inside and outside the curve). Also, the second term in ∇E_{image} has indeed the advantageous effect of better exploiting some of the information built in the initial contour (e.g., placement, shape or size) and allows for the segmentation of objects/structures that are in its vicinity. Furthermore, these experiments suggest that using statistical

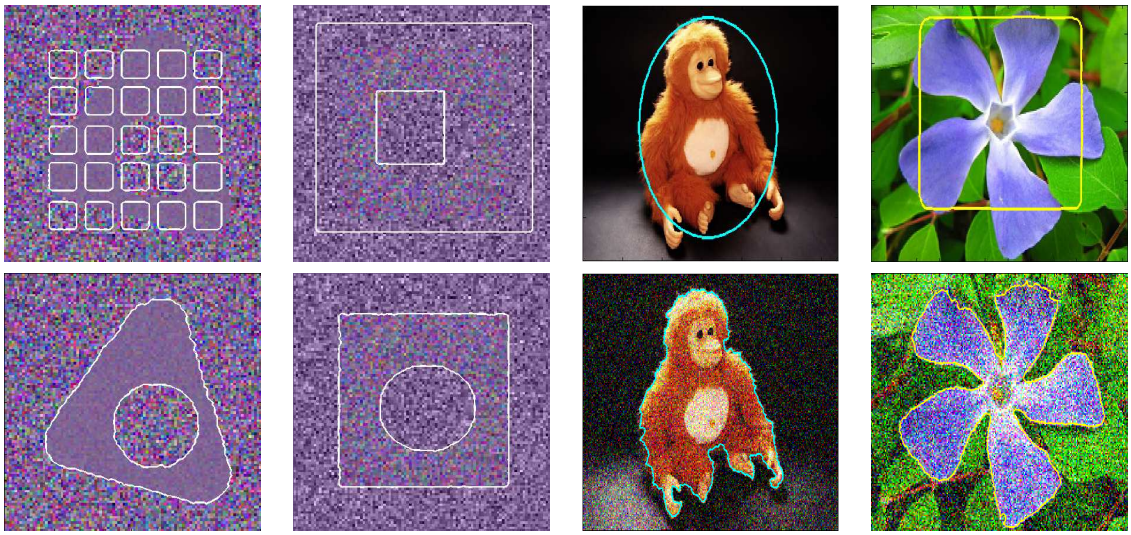


Figure 20: Segmentations of noisy images (see text). *First Line:* initialization; *Second line:* Results.

information alone may not be enough to lead to meaningful segmentation results.

3.5 Chapter Conclusion

In this chapter, we proposed a novel region-based segmentation technique with active contours. The method realizes a trade-off between the segmentation obtained from the segmenting curve and the segmentation obtained from thresholding the image (which is itself implied by the knowledge of the curve). Hence, our approach combines two segmentation techniques usually applied separately and allows us to take advantage of the power of the thresholding technique (various possible methods can be integrated in our framework) and the power of active contours (regularization, topological changes, localization...). An interesting feature of this work, is that we show that one can perform region-based segmentation based on geometric considerations of the image only. In particular, there is no need to fit particular statistical models to the image regions, which can be beneficial when such models poorly describe the real intensity distributions of regions and thus cannot lead to meaningful segmentations. Our framework is general enough, however, to allow the use of diverse

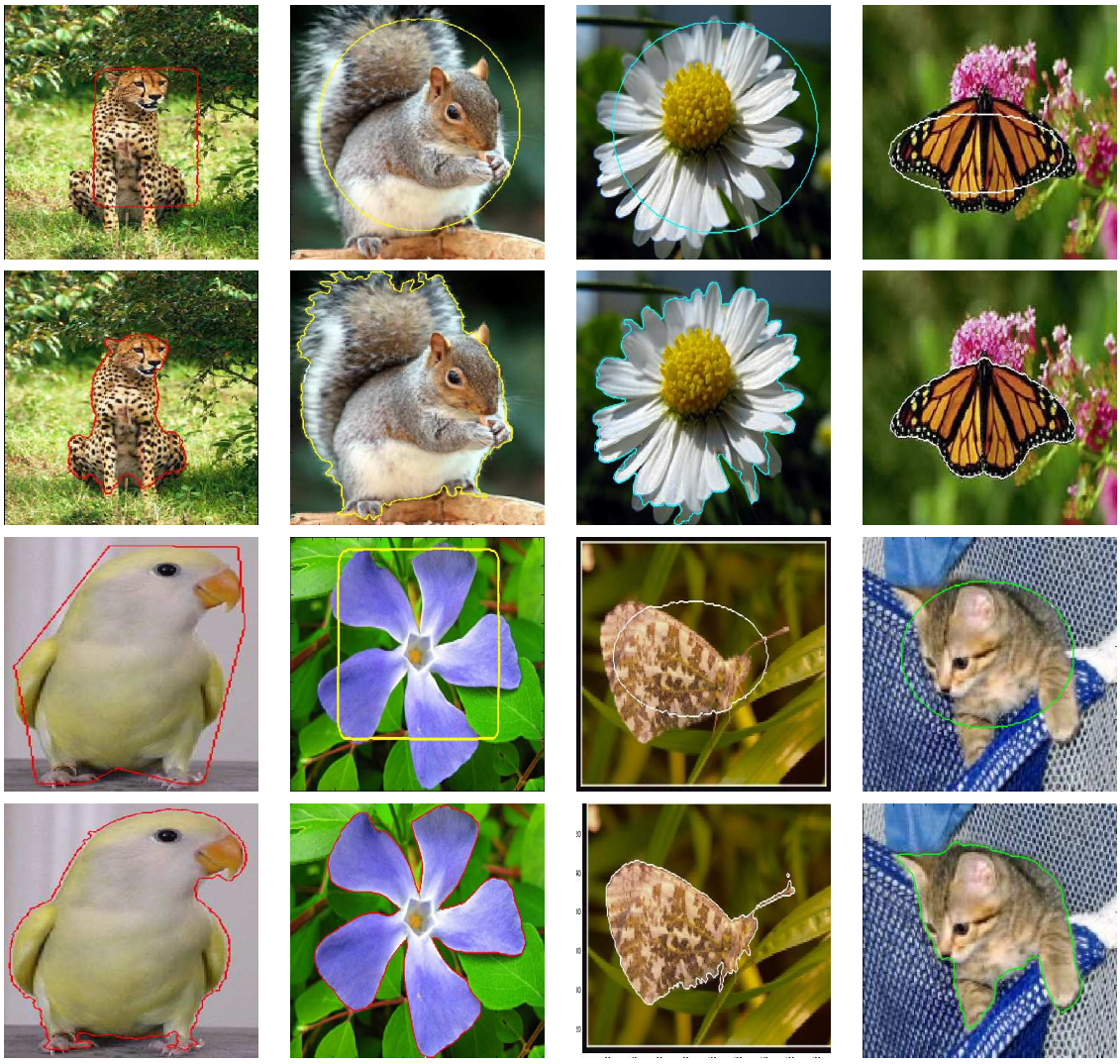


Figure 21: Segmentation results of natural images, in the presence of texture or complex background. *First and Third Rows:* Initializations; *Second and Fourth Rows:* Results.

statistical models. Interestingly, even when using statistical models, our flow has distinctive properties leading to meaningful segmentations on a wide range of nontrivial examples when compared to standard region-based approaches. These results stem from the shape interpretation of our energy functional, through which geometric information present in the image is also exploited in addition to statistical one. Hence, our methodology can alleviate the need for learning shape priors or for artificially imposing constraints on pixels statistics.

A possible extension and interesting avenue for future research could be to adapt the method to perform adaptive thresholding. Currently the method involves global thresholds that are constant over the image domain. Allowing the thresholds to be location dependant by defining a thresholding surface for instance (instead of a plane), would allow to segment an even larger class of images. One can expect that an essential step in devising an effective adaptive thresholding active contour approach will be to define a meaningful energy functional that regularizes the thresholding surface.

CHAPTER IV

NONLINEAR SHAPE PRIORS FROM KERNEL SPACE FOR GEOMETRIC ACTIVE CONTOURS

In this chapter, we present shape learning techniques based on kernel methods, for geometric active contours (GAC). This chapter details a few algorithms based on linear principal component analysis (LPCA), kernel PCA (KPCA), locally linear embedding (LLE) and kernel LLE (KLLE) that can be used to provide shape prior by learning a set of possible shapes, to improve segmentation performances with GACs. This chapter is organized as follows: In the first section, we provide some motivation and related work to the proposed approaches. In the second section, we present a consistent method to introduce shape priors within the GAC framework that uses the projections on the linear and kernel PCA feature spaces directly (feature space formulation). Using the proposed approach, we compare the performances of linear and kernel PCA, and demonstrate the superiority of kernel PCA on various challenging examples. This second section is based on [24] and part of [26, 28]. In the third section, we present methods to compute pre-images of projections in the KPCA and KLLE spaces (pre-image formulation), with a special focus on shape analysis. In this section, a comparison is made between LPCA, LLE, KLLE, and KPCA in terms of their ability to represent a previously unseen shape, using only the shapes available in the training data set. This third section is based on [99, 98].

4.1 Motivation and Related Work

Segmentation consists of extracting an object from an image, a ubiquitous task in computer vision applications. The active contour methodology has proven to be very effective for performing this task. However, the use of image information alone often leads to poor segmentation results in the presence of noise, clutter or occlusion. Furthermore, images are two dimensional projections of 3D space. This can result in ambiguities/distortions in the shape of an object as seen in 2D images. Other effects like camera or object motion and an imperfect camera sensor can also add to the distortions in the image, making segmentation a very challenging task. The introduction of shape priors in the active contour framework has been shown to be an effective way to address the aforementioned issues, leading to more robust segmentation (or tracking) performances. Introducing prior knowledge of shapes can be done using an appropriate shape learning technique applied on a given set of training shapes. However, the number of shapes in training sets is usually limited, and a main limitation of learning methods is the ability to accurately represent the possible variations in the shape of the object using only a few examples.

A number of methods that use a parameterized or an explicit representation for contours have been proposed [19, 129, 20] for active contour segmentation with shape priors. For instance, Cootes *et al.* [19] developed a parametric point distribution model for describing the segmenting curve by using linear combinations of eigenvectors that reflect variations from the mean shape. In [129], Wang and Staib developed a statistical point model by applying linear PCA to the covariance matrices that captures the statistical variations of the landmark points. In [21], the authors use the B-spline parametrization to build shape models in the kernel space [80]: The distribution of shapes in kernel space was assumed to be Gaussian and a Mahalanobis distance was minimized during the segmentation process to provide shape prior.

The geometric active contour framework (see [109, 89] and the references therein) involves a parameter-free representation of contours. Within the GAC framework, the authors in [70] obtain the shape statistics by performing linear principal component analysis (PCA) on a training set of signed distance functions (SDFs). This approach was shown to be able to convincingly capture small variations in the shape of an object. It inspired other schemes to obtain shape prior described in [124, 107], notably, where PCA were performed on SDFs to learn the shape variations.

However, when the object considered for learning undergoes complex or nonlinear deformations, linear PCA can lead to unrealistic shape priors, by allowing linear combinations of the learned shapes that are unfaithful to the true shape of the object. Cremers *et al.* [22], successfully pioneered the use of kernel methods to address this issue within the GAC framework, using a Parzen estimator [94] to model the shape distribution in kernel space.

In this chapter, we propose to use kernel PCA to introduce shape priors for GACs. Kernel PCA (KPCA) was proposed by Scholkopf *et al.* [80], and allows one to combine the precision of kernel methods with the reduction of dimension in training sets. Other machine learning methods such as Locally Linear Embedding (LLE) and Kernel LLE (KLLE) are also investigated as means to capture nonlinear dependencies in data-sets of shapes and to introduce better shape priors. This is the first time, to our knowledge, that KPCA, LLE, KLLE are explicitly used and compared as means to introduce shape priors within the GAC framework.

4.2 Shape Prior from Kernel PCA: Feature Space Formulation

In this section, we recall a general formulation allowing one to perform linear PCA as well as kernel PCA on any data set [113, 67]. Then we present specific kernels allowing to perform linear or nonlinear principal component analysis on training sets

of shapes. Finally, we propose an energy functional allowing to introduce shape priors obtained from either linear or kernel PCA, within the geometric active contour (GAC) framework. Although the present framework is general enough to use other learning methods, we focus on kernel PCA in what follows.

4.2.1 Kernel PCA

Kernel PCA (KPCA) can be considered to be a generalization of linear principal component analysis. This technique was introduced by Scholkopf [80], and has proven to be a powerful method to extract nonlinear structures from a data set. The idea behind KPCA consists of mapping a data set from an input space $\mathcal{I}$ into a feature space F via a nonlinear function φ. Then, PCA is performed in F to find the orthogonal directions (principal components) corresponding to the largest variation in the mapped data set. The first l principal components account for as much of the variance in the data as possible by using l directions. In addition, the error in representing any of the elements of the training set by its projection onto the first l principal components is minimal in the least squares sense.

The nonlinear map $\varphi : \mathcal{I} \mapsto F$ typically does not need to be known, through the use of Mercer kernels. A *Mercer kernel* is a function $k(\cdot, \cdot)$ such that for all data points χ_i, the kernel matrix

$$\mathbf{K} = \begin{pmatrix} k(\chi_1, \chi_1) & k(\chi_1, \chi_2) & \dots & k(\chi_1, \chi_N) \\ k(\chi_2, \chi_1) & \ddots & & \\ \vdots & & k(\chi_i, \chi_j) & \\ k(\chi_N, \chi_1) & & & k(\chi_N, \chi_N) \end{pmatrix} \tag{87}$$

is symmetric positive [80]. According to Mercer's Theorem (see [77]), computing $k(.,.)$ as a function of $\mathcal{I} \times \mathcal{I}$, amounts to computing the inner scalar product in F: $k(\chi_a, \chi_b) = \langle \varphi(\chi_a), \varphi(\chi_b) \rangle$, with $(\chi_a, \chi_b) \in \mathcal{I} \times \mathcal{I}$. This scalar product in F defines a distance d_F, such as $d_F^2(\varphi(\chi_a), \varphi(\chi_b)) = \| \varphi(\chi_a) - \varphi(\chi_b) \|^2 = k(\chi_a, \chi_a) - 2k(\chi_a, \chi_b) + k(\chi_b, \chi_b)$.

We now describe the KPCA method [80]. Let $\mathcal{T} = \{\chi_1, \chi_2, ..., \chi_N\}$ be a set of training data. The centered kernel matrix $\tilde{\mathbf{K}}$ corresponding to $\mathcal{T}$, is defined as

$$\begin{aligned} \tilde{\mathbf{K}}(i,j) &= \langle (\varphi(\chi_i) - \bar{\varphi}), (\varphi(\chi_j) - \bar{\varphi}) \rangle \\ &= \langle \tilde{\varphi}(\chi_i), \tilde{\varphi}(\chi_j) \rangle = \tilde{k}(\chi_i, \chi_j), \quad \text{for } (i,j) \in [|1, N|] \end{aligned} \tag{88}$$

with

$$\bar{\varphi} = \frac{1}{N} \sum_{i=1}^{N} \varphi(\chi_i) \ ,$$

where

$$\tilde{\varphi}(\chi_i) = \varphi(\chi_i) - \bar{\varphi}$$

is the centered map corresponding to χ_i, and $\tilde{k}(.,.)$ denotes the centered kernel function.

Since $\tilde{\mathbf{K}}$ is symmetric, it can be decomposed as

$$\tilde{\mathbf{K}} = \mathbf{U}\mathbf{S}\mathbf{U}^T \tag{89}$$

where $\mathbf{S} = \text{diag}(\gamma_1, ..., \gamma_N)$ is a diagonal matrix containing the eigenvalues of $\tilde{\mathbf{K}}$. $\mathbf{U} = [\mathbf{u}_1, ..., \mathbf{u}_N]$ is an orthonormal matrix. The column-vectors $\mathbf{u}_i = [u_{i1}, ..., u_{iN}]^T$ are the eigenvectors corresponding to the eigenvalues γ_i's. Furthermore, it can easily be shown that

$$\tilde{\mathbf{K}} = \mathbf{H}\mathbf{K}\mathbf{H}$$

where $\mathbf{H} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$. $\mathbf{1} = [1, ..., 1]^T$ is an $N \times 1$ vector.

Let $\mathbf{C}$ denote the covariance matrix of the elements of the training set mapped by $\tilde{\varphi}$. Within the KPCA methodology, the covariance matrix $\mathbf{C}$, which is possibly of very high dimension, does not need to be computed explicitly. Only $\tilde{\mathbf{K}}$ needs to be known to extract features from the training set, since the eigenvectors of $\mathbf{C}$ are simple functions of the eigenvectors of $\tilde{\mathbf{K}}$. Indeed, denoting by V_k the k^{th} (orthonormal) eigenvector of the covariance matrix in the feature space, one has (see e.g., [113])

$$V_k = \sum_{i=1}^{N} \frac{u_{ki}}{\sqrt{\gamma_k}} \tilde{\varphi}(\mathbf{x}_i).$$

The subspace of the feature space F spanned by the first l eigenvectors of $\mathbf{C}$, will be referred to as the *kernel PCA space* or *KPCA space*, in what follows: The kernel PCA space is the subspace of F, obtained from learning the training data.

Let χ be *any* element of the input space $\mathcal{I}$. The projection of χ on the KPCA space will be denoted by $P^l\varphi(\chi)$ [1]. Let β_k denote the coordinate of $P^l\varphi(\chi)$ on the k^{th} component V_k, one has (see e.g., [80])

$$\beta_k = \frac{1}{\sqrt{\gamma_k}} \sum_{i=1}^{N} u_{ki} \tilde{k}(\chi, \chi_i). \tag{90}$$

The projection of $\varphi(\chi)$ onto the subspace spanned by the first l eigenvectors is then given by

$$P\varphi^l(\chi) = \sum_{k=1}^{l} \beta_k V_k + \bar{\varphi} \tag{91}$$

The squared distance d_F^2 between $\varphi(\chi)$ and its projection on the KPCA space is given by

$$\begin{aligned} d_F^2[\varphi(\chi), P^l\varphi(\chi)] &= \| \varphi(\chi) - P^l\varphi(\chi) \|^2 \\ &= k(\chi,\chi) - 2\langle \varphi(\chi), P^l\varphi(\chi) \rangle + \langle P^l\varphi(\chi), P^l\varphi(\chi) \rangle \end{aligned} \tag{92}$$

This distance measures the discrepancy between a mapped element $\varphi(\chi)$ and the learnt space, and will be minimized to introduce shape knowledge in the contour evolution process, in Section 4.2.3. Using some matrix manipulations, this squared distance can be expressed only in terms of kernels as

$$d_F^2[\varphi(\chi), P^l\varphi(\chi)] = k(\chi,\chi) + \frac{1}{N^2}\mathbf{1}^T\mathbf{K}\mathbf{1} - \frac{2}{N}\mathbf{1}^T\mathbf{k}_\chi + \tilde{\mathbf{k}}_\chi^T\mathbf{M}\tilde{\mathbf{K}}\mathbf{M}\tilde{\mathbf{k}}_\chi - 2\tilde{\mathbf{k}}_\chi^T\mathbf{M}\tilde{\mathbf{k}}_\chi \tag{93}$$

where, $\tilde{\mathbf{k}}_\chi = \mathbf{H}(\mathbf{k}_\chi - \frac{1}{N}\mathbf{K}\mathbf{1})$ and $\mathbf{M} = \sum_{i=1}^{l} \frac{1}{\gamma_i}\mathbf{u}_i\,\mathbf{u}_i^T$.

Also, the squared distance between the projection $P^l\varphi(\chi)$ and a specific (mapped) element of the training set $\varphi(\chi_i)$ can be written as (see e.g., [68])

$$\begin{aligned} d_F^2[\varphi(\chi_i), P^l\varphi(\chi)] &= \| \varphi(\chi_i) - P^l\varphi(\chi) \|^2 \\ &= \| P^l\varphi(\chi) \|^2 + \| \varphi(\chi_i) \|^2 - 2\langle P^l\varphi(\chi), \varphi(\chi_i) \rangle \end{aligned} \tag{94}$$

[1] In this notation, l refers to the first l eigenvectors of $\mathbf{C}$ used to build the KPCA space.

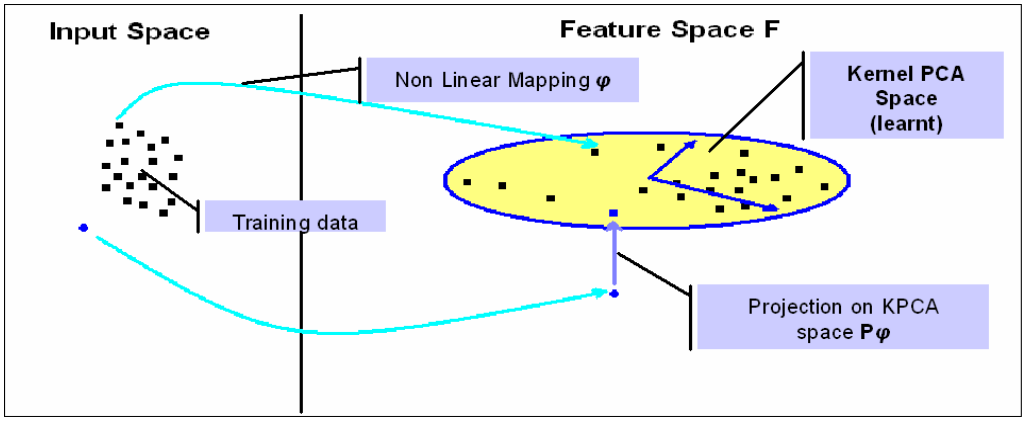


Figure 22: Kernel PCA methodology. A training set is mapped from input space $\mathcal{I}$ to feature space F, via a nonlinear function φ. PCA is performed in F to determine the principal directions defining the kernel PCA space (learned space): Oval area. Any element of $\mathcal{I}$ can then be mapped to F and projected on the kernel PCA space via $P^l\varphi$.

After some matrix manipulations, this squared distance can be written only in terms of the kernel as

$$d_F^2[\varphi(\chi_i), P^l\varphi(\chi)] = \left(\mathbf{k}_\chi + \frac{1}{N}\mathbf{K1} - 2\mathbf{k}_{\chi_i}\right)^T \mathbf{H}^T\mathbf{M}\mathbf{H}\left(\mathbf{k}_\chi - \frac{1}{N}\mathbf{K1}\right) + \frac{1}{N^2}\mathbf{1}^T\mathbf{K1} + K_{ii} - \frac{2}{N}\mathbf{1}^T\mathbf{k}_{\chi_i} \tag{95}$$

where and $K_{ii} = k(\chi_i, \chi_i)$.

Figure 22 recapitulates the kernel PCA methodology as well as the projection operation on the learned space.

4.2.2 Kernels for linear and nonlinear PCA

We now present a few popular kernels that allow to perform learning of shapes using linear and nonlinear PCA.

4.2.2.1 Linear PCA

In [70], a method is presented to learn shape variations by performing PCA on a training set of shapes (closed curves) represented as the zero level sets of signed

distance functions [116, 87]. Using the following kernel in the formulation of kernel PCA presented above amounts to performing linear PCA on the given SDFs:

$$k_{id}(\phi_i, \phi_j) = \langle \phi_i, \phi_j \rangle = \int \int \phi_i(u, v)\phi_j(u, v)du.dv \tag{96}$$

for all SDFs ϕ_i and $\phi_j : \mathbf{R}^2 \mapsto \mathbf{R}$. The subscript id stands for the identity function: when performing linear PCA the kernel used is the inner scalar product in the input space, hence the corresponding mapping function $\varphi = id$.

A different representation for shapes is based on the use of binary maps, i.e., one sets to 1 the pixels located inside the shape, and to 0 the pixels located outside (see Figure 23). One can change the shape representation from SDFs to binary maps using the Heaviside function $H(\phi) = H\phi = \begin{cases} 1 & \text{if} \quad \phi \geq 0 \ , \\ 0 & \text{else} \ . \end{cases}$. Note that, in this case, the kernel allowing one to perform linear PCA is given by

$$k_{id}(H\phi_i, H\phi_j) = \langle H\phi_i, H\phi_j \rangle. \tag{97}$$

In numerical applications, a smooth version $H_\epsilon\phi$ of $H\phi$ can be obtained by taking $H_\epsilon\phi = \frac{1}{2} + \frac{1}{\pi}\arctan(\frac{\phi}{\epsilon})$, for ϵ small. The derivative of H will be noted δ, in the rest of this chapter. In numerical applications, a smooth version $\delta_\epsilon\phi$ of $\delta\phi$ can be obtained by taking $\delta_\epsilon\phi = \frac{1}{\pi}\frac{1}{1+(\frac{\phi}{\epsilon})^2}$.

4.2.2.2 Nonlinear PCA

Choosing a nonlinear kernel function $k(\cdot, \cdot)$ is the basis of nonlinear PCA. The exponential kernel has been a popular choice in the machine learning community and has proven to nicely extract nonlinear structures from data sets; see e.g. [67]. Using SDFs for representing shapes, this kernel is given by

$$k_{\varphi_\sigma}(\phi_i, \phi_j) = e^{-\frac{\|\phi_i - \phi_j\|^2}{2\sigma^2}}, \tag{98}$$

where σ^2 is a variance parameter estimated *a priori* and $\|\phi_i - \phi_j\|^2$ is the squared L_2-distance between two SDFs ϕ_i and ϕ_j. The subscript φ_σ stands for the nonlinear mapping corresponding to the exponential kernel; this mapping also depends on the choice of σ. If the shapes are represented by binary maps, the corresponding kernel is

$$k_{\varphi_\sigma}(H\phi_i, H\phi_j) = e^{-\frac{\|H\phi_i - H\phi_j\|^2}{2\sigma^2}}. \tag{99}$$

This exponential kernel is one among many possible choices of Mercer kernels. Other kernels may be used to extract other specific features from the training set; see [80]. For instance, another widely used kernel in the literature is the polynomial kernel given by

$$k_{\varphi_P}(\phi_i, \phi_j) = (c + \langle \phi_i, \phi_j \rangle)^d \tag{100}$$

where c is any constant and d is the degree (odd) of the polynomial. For shapes represented as binary maps, the polynomial kernel becomes

$$k_{\varphi_P}(H\phi_i, H\phi_j) = (c + \langle H\phi_i, H\phi_j \rangle)^d \tag{101}$$

4.2.3 Shape Prior for GAC from KPCA Feature Space

To include prior shape knowledge in the GAC framework, we propose to use the projection on the kernel PCA space as a model and to minimize the following energy functional:

$$E^{\mathrm{F}}_{\mathrm{shape}}(\chi) := d^2_F[\varphi(\chi), P^l\varphi(\chi)]. \tag{102}$$

The superscript F in $E^{\mathrm{F}}_{\mathrm{shape}}$ denotes the fact that the shape knowledge is expressed as a distance in feature space. A similar idea was presented in [114], for the purpose of pattern recognition and denoising. In Equation (102), χ is a test shape represented using either a SDF ($\chi = \phi$) or a binary map ($\chi = H\phi$) and φ refers to either id (linear PCA) or φ_σ (kernel PCA). Minimizing $E^{\mathrm{F}}_{\mathrm{shape}}$ as in Equation (12) amounts to driving the test shape χ towards the kernel PCA space computed a priori from a training

set of shapes using (89). This contour evolution involving only the minimization of $E^{\mathrm{F}}_{\mathrm{shape}}$ (no image information) will be referred to as "warping," in what follows.

A number of researchers have proposed minimizing the distance between the current shape and the mean shape obtained from a training set. The assumption is indeed often made that the underlying distribution of familiar "shapes," in either the input or feature space, is Gaussian [70, 20, 21]. Following this assumption, driving the curve toward the mean shape is a sensitive choice. Here, however, we deliberately chose to use the projection of the (mapped) current SDF to drive the evolution because we would like to deal with objects of different geometry in the training set (see e.g., Figure 23, third line; a training set of 4 words was used for the experiments). When dealing with objects of very different shapes, the underlying distribution can be quite non-Gaussian (e.g., multi-modal).

Thus, the average shape would not be meaningful in this case, since it would amount to mixing shapes belonging to different clusters. As a consequence, driving the (mapped) current shape towards its projection on the kernel PCA space appears to be a more sensible choice for our purposes. In addition, choosing the projection as a model of shape allows for comparing the given learning methods without image information, i.e., by warping the same initial contour for each method and comparing the final results in terms of their resemblance to the elements of the training set. The final shape obtained can be interpreted as "the most probable shape of the initial contour given what is known from the data" (see the warping experiments in Section 4.2.4). If the mean shape is chosen as a shape model, warping would result in the initial contour converging to this mean shape, and no comparison would be possible among learning methods. Thus, in this latest case, image information must be included and it is difficult to conclude whether differences between segmentation results are due to differences in the performances of the learning methods or simply to poor balancing between image and shape information.

The gradient of E_{shape} can be computed by applying the calculus of variations in the definition of the energy given in Equation (102), using the expression of $d_F^2(\varphi(\xi), P^l\varphi(\xi))$ in Equation (93). The energy $E^{\text{F}}_{\text{shape}}(\chi)$ can be minimized as follows:

$$\frac{d\phi}{dt} = -\nabla_\phi E^{\text{F}}_{\text{shape}} = -\nabla_\chi E^{\text{F}}_{\text{shape}}.\frac{d\chi}{d\phi}. \tag{103}$$

The minimization of E_{shape} for any arbitrary contour (no image information) results in the deformation of the contour towards a familiar shape (as presented in Section 4.2.4).

For the exponential kernel involving SDFs and given in Equation (98), the following result is obtained:

$$\nabla_\phi E^{\text{F}}_{\text{shape}} = -\frac{\Sigma_{i=1}^N g_i(\phi).k_{\varphi_\sigma}(\phi,\phi_i).[\phi-\phi_i]}{\sigma^2} \tag{104}$$

with $[g_1(\phi),...,g_N(\phi)] = -\frac{2}{N}\mathbf{1}^T + 2\tilde{\mathbf{k}}_\phi^T\mathbf{M}\tilde{\mathbf{K}}\mathbf{M}\mathbf{H} - 4\tilde{\mathbf{k}}_\phi^T\mathbf{M}\mathbf{H}$, and $\tilde{\mathbf{k}}_\phi$, $\mathbf{M}$ and $\tilde{\mathbf{K}}$ computed for k_{φ_σ}.

For the exponential kernel involving binary maps and given in Equation (99), one derives that

$$\nabla_\phi E^{\text{F}}_{\text{shape}} = -\frac{\Sigma_{i=1}^N g_i(\phi).k_{\varphi_\sigma}(H\phi,H\phi_i).\delta\phi.[H\phi-H\phi_i]}{\sigma^2} \tag{105}$$

where $\tilde{\mathbf{k}}_\phi$, $\mathbf{M}$ and $\tilde{\mathbf{K}}$ are computed for the kernel $k_{\varphi_\sigma}(H.,H.)$.

For the kernel given in Equation (96), corresponding to linear PCA on SDFs, the following result is obtained:

$$\nabla_\phi E^{\text{linear}}_{\text{shape}} = 2\phi + \Sigma_{i=1}^N g_i(\phi).\phi_i \tag{106}$$

where $\tilde{\mathbf{k}}_\phi$, $\mathbf{M}$ and $\tilde{\mathbf{K}}$ are computed for the kernel k_{id}.

Finally, for the kernel given in Equation (97), corresponding to linear PCA on binary maps, one finds that

$$\nabla_\phi E^{\text{linear}}_{\text{shape}} = 2\delta\phi + \Sigma_{i=1}^N g_i(H\phi).H\phi_i.\delta\phi \tag{107}$$

where $\tilde{\mathbf{k}}_\phi$, $\mathbf{M}$ and $\tilde{\mathbf{K}}$ are computed for the kernel $k_{id}(H., H.)$.

4.2.4 Warping Experiments: Linear PCA vs KPCA for shape priors

In this section, we compare the performances of kernel PCA to linear PCA, as methods for introducing shape priors in the GAC framework. The fact that energy functional E^F_{shape} is consistently defined for both shape learning methods in Equation (102), ensures that performances obtained from applying linear PCA or KPCA can be accurately compared. No image information was used in these experiments, instead the contour evolution was carried to minimize E^F_{shape} only (warping). This further guarantees a meaningful comparison among methods, since performances do not stem from the balancing factor between image information and shape knowledge. In what follows, the same initial shapes were warped and the final contours obtained were compared, for both methods, in terms of their resemblance to the elements of the training sets utilized. The equation

$$\frac{\partial \phi}{\partial t} = -\nabla_\phi E^F_{\text{shape}} \tag{108}$$

was run until convergence, using the expression of the gradients presented in (106) and (107) for linear PCA, and (104) and (105) for KPCA.

4.2.4.1 Training sets and learning

In the experiments presented below, three training sets of shapes were used. Shapes were represented alternatively by signed distance functions and (smoothed) binary maps. The first training set of shapes consists of 22 shapes of a man playing soccer. The second training set is composed of 28 shapes of a shark. These shapes were aligned using an appropriate registration scheme [124] to remove differences between them resulting from translation, rotation, and scale. To test for the ability of the proposed framework to learn and deal with multi-modal distributions, a third training set was

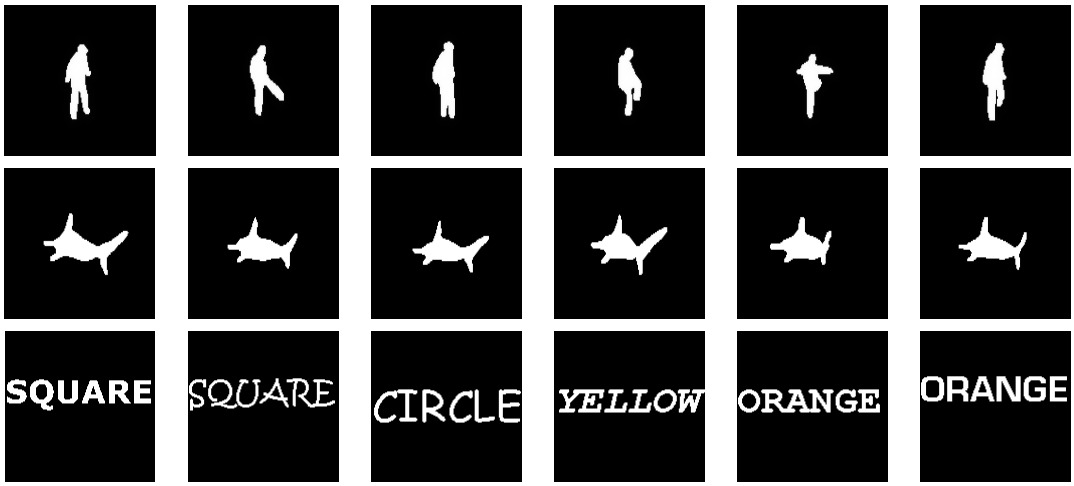


Figure 23: Three training sets (Before alignment - Binary images are presented here). First row, Soccer Player Training Set (6 of the 22 used). Second row, Shark Training Set (6 of the 15 used). Third row, 4-Words Training Set (6 of the 80 learned; 20 fonts per word).

built. This training set consists of four words, *orange*, *yellow*, *square* and *circle* each written using 20 different fonts, leading to a training set of 80 shapes total. The size of the fonts was chosen to lead to words of roughly the same length. The obtained words were then registered according to their centroid. No further effort, such as matching the letters of the different words, was pursued. The binary maps corresponding to the diverse training shapes are presented in Figure 23. The shapes depicted are from the original training sets, before alignment. The first row in the figure presents a few elements of the Soccer Player Training Set, and the second row presents a few shapes from the Shark Training Set. The third row of Figure 23 presents a few of the words used to build the 4-Words Training Set. Shape learning was performed on each training set, as presented in Section 4.2. The familiar spaces of shapes (kernel PCA spaces) were built for each of the kernels presented in Equations (96) to (99), whether linear PCA or KPCA was performed on binary maps or SDFs.

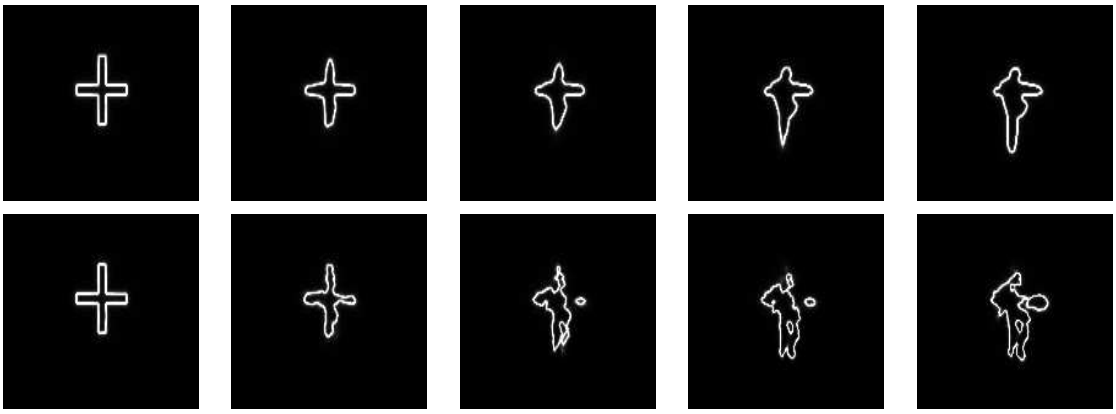
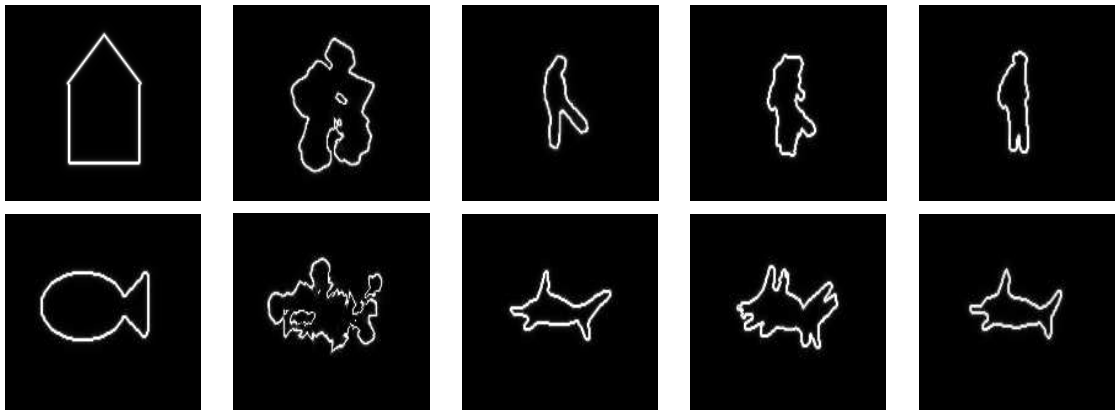


Figure 24: Warping results obtained for the Soccer Player Training Set, starting with an initial contour representing a cross. First line: Evolution obtained using kernel PCA; Second line: Evolution obtained using linear PCA.

4.2.4.2 Warping an arbitrary initial shape

Figure 24 presents the warping results obtained for an arbitrary initial shape representing a cross. The first row of the figure shows the results obtained using kernel PCA (with SDFs as a representation of shape). The second row of the figure presents the results obtained using linear PCA (with shapes represented as SDFs). As can be noticed, results obtained with linear PCA bear little resemblance to the elements of the training sets. By contrast, final contours obtained employing KPCA are more faithful to the learned shapes.

Figure 25(b) shows the warping results obtained by applying linear PCA on SDF. Figure 25(d) shows the warping results obtained by applying linear PCA on binary maps. Note that the results obtained for the SDF representation bear little resemblance to the elements of the training sets. Results obtained for binary maps are more faithful to the learned shapes. Figure 25(c) and (e) present the warping results obtained by applying kernel PCA on SDF and binary maps, respectively. In both cases, the final contour is very close to the training set and results are better than any of the results obtained with linear PCA.



(a) (b) (c) (d) (e)

Figure 25: Warping results of an arbitrary shape, obtained using linear PCA and kernel PCA applied on both signed distance functions and binary maps. First row: Results for the Soccer Player Training Set, Second row: Results for the Shark Training Set. (a): Initial shape, (b): PCA on SDF, (c): kernel PCA on SDF (d): PCA on binary maps, (e): kernel PCA on binary maps.

4.2.4.3 Robustness to misalignment

In these experiments, the robustness of each method (linear and kernel PCA) to misalignments relative to the registered shapes of a training set is tested. This robustness to misalignment is an interesting property to evaluate since, to perform segmentation, differences in pose (e.g., translations) between the object of interest in the image and the registered training shapes need to be accounted for. Hence, whether a probabilistic method [70] or a gradient descent scheme [124] is used to compensate for these differences, more or less important misalignments between the contour and the registered training shapes may occur during the evolution process. This can impair the ability of the shape energy to properly constrain the shape of the segmenting contour.

In each of the following three experiments, an initial contour representing a chosen training shape for which pose differs from the registered training shapes is used as the initial contour for warping (SDFs were used to represent shapes). For the first initial contour, a small translation of about 5 pixels in the x and y-directions was performed. The second initial contour was translated of about 10 pixels in the x

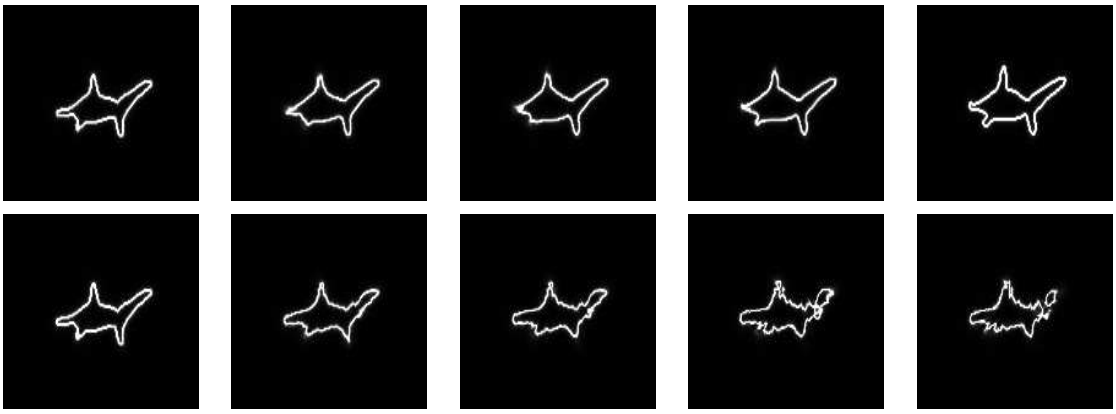


Figure 26: Warping results obtained for the Shark Training Set. The initial contour (leftmost image) represents one of the learned shapes slightly misaligned (5 pixels) with the corresponding element of the registered training set. First line: Evolution obtained using kernel PCA; Second line: Evolution obtained using linear PCA.

and y-directions. Figure 26 and Figure 27 present the warping results obtained for the smaller and larger misaligned initial shapes, respectively. The first line of each of these figures shows the results obtained using kernel PCA and the second line presents the results using linear PCA. Once again, results obtained with linear PCA bear little resemblance to the elements of the training sets. By contrast, final contours obtained using kernel PCA are more faithful to the learned shapes. In addition, whether the initial contour is slightly or heavily misaligned, the contour evolution results in a translation of the contour when kernel PCA is used. Interestingly, the shape of the final contour obtained using KPCA is very much like the shape of the initial contour, for both experiments. In Figure 28, the initial contour represents one of the learned shapes scaled up of 10% and rotated of 10°. As a result of warping using linear PCA, an unrealistic shape is obtained. In contrast, the final contour obtained when using KPCA is a realistic shape that is in effect rotated back and scaled down compared with the initial contour. Similar satisfying results were obtained with kernel PCA using binary maps as a representation of shapes. Hence, KPCA appears to offer a higher level of robustness than linear PCA to misalignment of the initial contour (at least in the experiments we performed).

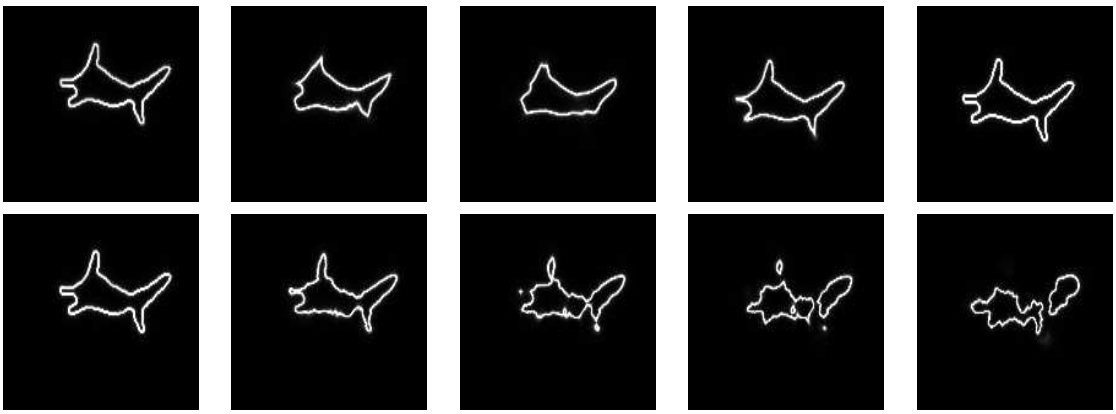


Figure 27: Warping results obtained for the Shark Training Set. The initial contour (leftmost image) represents one of the learned shapes misaligned of approximately 15 pixels. First line: Evolution obtained using kernel PCA; Second line: Evolution obtained using linear PCA.

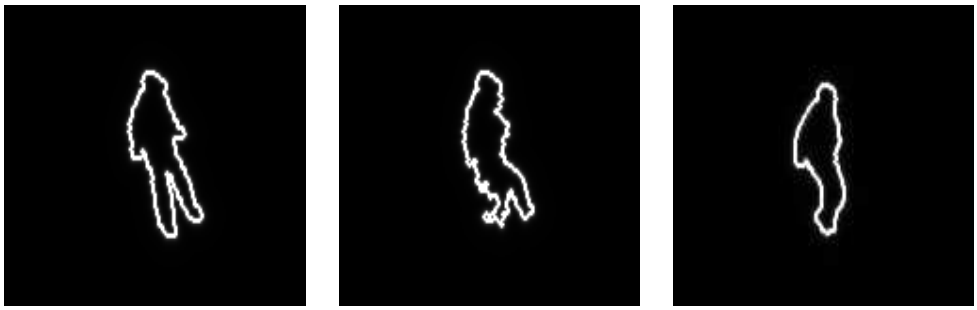


Figure 28: Warping results obtained for the "Soccer player" training set. *Left:* Initial contour representing one of the learned shapes scaled up of 10% and rotated of 10°; *middle:* Final result using linear PCA (an unrealistic shape is obtained); *right:* Final result using KPCA (Final contour is a realistic shape, effectively rotated and scaled down).

4.2.4.4 Multi-modal learning

Kernel methods have been used to learn complex multi-modal distributions in an unsupervised fashion [80]. The goal of this section is to investigate the ability of kernel PCA to simultaneously learn objects of different shapes and to constrain the contour evolution in a meaningful fashion. Moreover, we want to contrast performances obtained with KPCA to performances obtained with linear PCA. The 4-Words Training Set was used for these experiments. Diverse contours, whose shapes bore some degree of resemblance to any of the four words (*orange*, *yellow*, *square* or *circle*), were then used as initial contours for warping.

Each line in Figure 29 presents the warping results obtained for the diverse initial contours used, for both linear and KPCA, using SDFs as representations of shape. The initial contour presented on the first line of Figure 29 is the word *square*, in which letters are partially erased. The warping result using linear PCA bears little resemblance to the word *square*. By contrast, using kernel PCA, the word is accurately reconstructed. In particular, a font close to the original one used in the initial contour is obtained. The initial contour presented on the second line of Figure 29 is the word *circle*, occluded by a line. Again, the warping result using linear PCA bears little resemblance to the word *circle*. In fact, the obtained result appears as a mixing of different words. Using KPCA, the word *circle* is not only accurately reconstructed but the line is completely removed. Moreover, the original font used for the initial contour (which belongs to the training set) is preserved. The initial contour presented on the third line of Figure 29 is one of the registered training words *yellow*, slightly translated. The letter "y" is also replaced by a rectangle. Employing kernel PCA, the word *yellow* is perfectly reconstructed. In particular, the letter "y" is recovered, the contour is translated back and the original font used for the initial contour is preserved. In comparison, the word *yellow* is barely recognizable from the final contour obtained

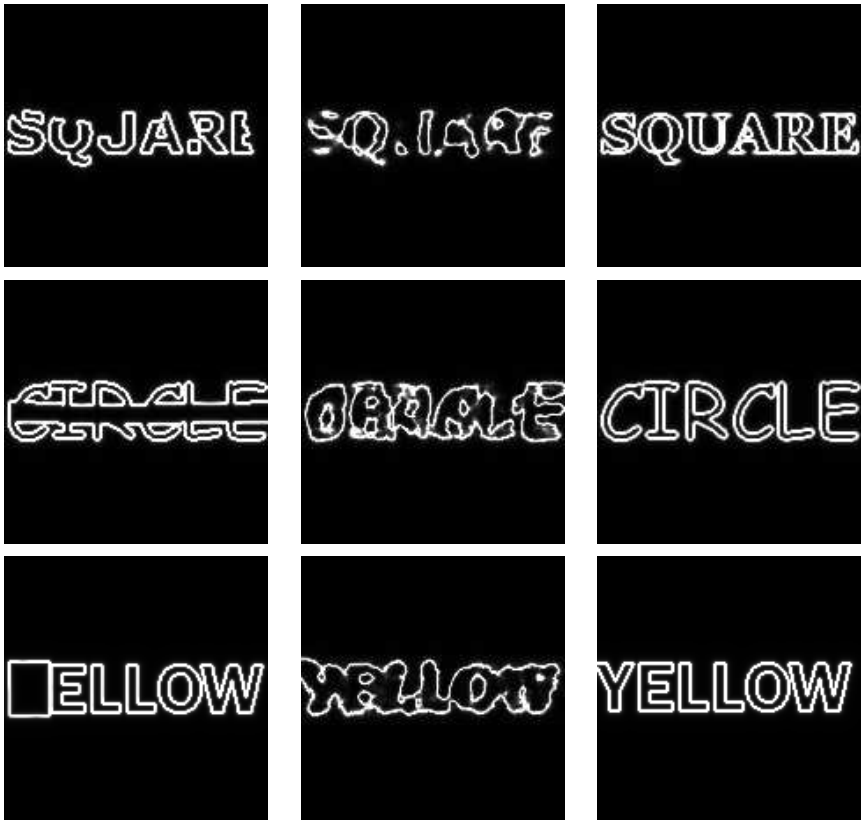


Figure 29: Warping Results for the 4-Words Training Set. *Left column*: Initial contours; *Middle Column*: Warping using linear PCA; *Right Column*: Warping using kernel PCA.

using linear PCA.

In each of the experiments above, the accurate word (i.e., closest to the word used to build the initial contour) is detected and reconstructed, using kernel PCA. Similar robust performance was obtained with kernel PCA on binary maps. The shape of the final contour obtained with linear PCA, was oftentimes the result of a mixing of words belonging to different classes. This mixing of classes can lead to unrealistic shapes. Thus, KPCA appears to be a strong method for introducing shape priors within the GAC framework, when training sets involving different types of shapes are used.

(a) (b) (c) (d) (f)

Figure 30: Influence of σ for the kernel PCA method (exponential kernel) applied on binary maps. Warping results of an arbitrary shape are presented for the Shark Training Set. (a): Initial shape, (b): Warping result for $\sigma = 3$, (c): $\sigma = 7$, (d): $\sigma = 9$, (e): $\sigma = 15$.

4.2.4.5 Influence of the parameter σ in exponential Kernels

The goal of this section is to study the influence of the parameter σ, when kernel PCA is performed using exponential kernels. Figure 30 presents warping results of an arbitrary shape using different values of the parameter σ for the kernel defined in Equation (99), which corresponds to performing kernel PCA on binary maps. The Shark training set was used for the shape learning.

As can be noticed in Figure 30, as the value of σ increases more and more mixing among the shapes of the training set is allowed. Such shape mixing naturally occurs, when linear PCA is used for learning (e.g., in Figure 25, second row (d), this type of mixing can be observed). Similar results were obtained for the SDF representation and the kernel given in Equation (98). Hence, the parameter σ allows for *controlling* the degree of mixing allowed among the learned shapes in the shape prior, which is another advantage of kernel PCA over linear PCA. The choice of σ should typically depend upon how much shape variation occurs within the data set. To emphasize, larger σ's allow for more mixing among the shapes, whereas smaller σ's lead to more selective shape priors.

4.3 Shape Prior from Kernel Methods: Pre-Image Formulation

4.3.1 Computing Pre-Images for Kernel PCA

4.3.1.1 The Pre-Image Problem

As alluded to above, the basic idea behind kernel methods is to map the data from the input space $\chi \in \mathcal{I}$ into a feature space F via some nonlinear map φ, and then apply a linear method in F to do further analysis. While the mapping φ from input space to feature space is of primary importance in kernel based methods, the reverse mapping from the feature space to input space can also be quite useful. For example, in image (or signal) de-noising, a given set of noisy images can be de-noised by applying kernel PCA [80]. The problem however is that the de-noised images so obtained are in the feature space and not in the input space. For visualizing the results or operating on them, obtaining the pre-image of the projection of a feature vector given in F (in the input space $\mathcal{I}$) can be critical. In this section, we use kernel PCA for doing statistical shape analysis on a set of shapes embedded in a signed distance function [70]. Knowing the pre-image can be quite useful in this case to visualize the different modes of variation of the shapes in the training set. It can also be used to provide shape prior for various tracking and segmentation tasks.

4.3.1.2 Prior Work addressing this Problem

However, as demonstrated by Mika [80], the exact pre-image typically does not exist and oftentimes one can only settle for an approximate solution. Yet, even finding approximate solutions may be non-trivial as the dimensionality of the feature space can be infinite. For certain invertible kernels, this (nonlinear) problem can be solved using a fixed-point iteration method as proposed by Scholkopf and Mika [80, 113].

However, this method is dependent on the initial starting point and is highly susceptible to local minima. To circumvent this problem, a new algorithm [68] was proposed to uniquely reconstruct the pre-image by utilizing distances of vectors in feature space and its relation to multi-dimensional scaling (MDS): To reconstruct the pre-image of a test vector $\chi \in \mathcal{I}$, the authors first find the distance of this point to all the training points in the feature space. Then, they find a new coordinate system to represent the data in the input space by performing SVD (singular value decomposition) on a matrix of n nearest neighbors. Next, they use MDS to project the solution into this new coordinate system. Finally, the eigenvectors of this new coordinate system are used to find the approximate pre-image in the original input space. Even though the method uses a couple of approximations for reconstruction, the results reported by the authors are impressive.

4.3.1.3 Proposed Approach/Contribution

In the present work, we formulate the problem so that the reconstruction algorithm does not use MDS and only uses the distances in the feature space to find an approximate pre-image. The proposed method performs only one approximation in the feature space and reduces computation time, as compared to [67]. We present the resulting algorithm for both the exponential and the polynomial kernels presented in Section 4.2.2.2, and use it to perform shape analysis and learning. Another comparison with linear PCA [70] confirms the advantages of using pre-images obtained from kernel PCA for shape analysis. The proposed approach supposes that invertible kernels of the form

$$k(\chi_i, \chi_j) = f\left(d^2_{\mathcal{I}}(\chi_i, \chi_j)\right)$$

are used in the learning phase, with $f : \mathbb{R}^+ \mapsto \mathbb{R}^+$ invertible, inverse f^{-1}, and $d^2_{\mathcal{I}}(\chi_i, \chi_j)$ the distance in input space. For most commonly used invertible kernels, the simple relationship between the input space distance $d^2_{\mathcal{I}}(\chi_i, \chi_j)$ and the feature

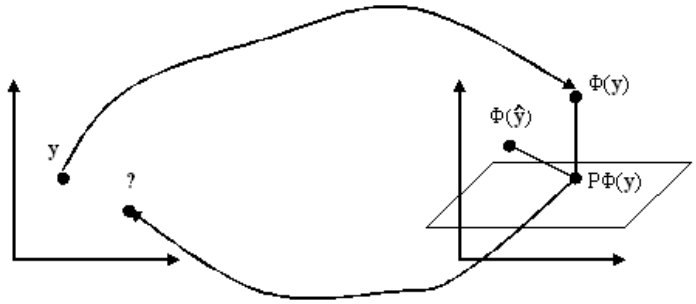


Figure 31: The pre-image problem in kernel PCA

space distance $d_F^2(\varphi(\chi_i), \varphi(\chi_j))$ holds [112]:

$$\begin{aligned} d_F^2(\varphi(\chi_i), \varphi(\chi_j)) &= \| \varphi(\chi_i) - \varphi(\chi_j) \|^2 \\ &= k(\chi_i, \chi_i) + k(\chi_j, \chi_j) - 2k(\chi_i, \chi_j) \\ &= K_{ii} + K_{jj} - 2f\left(d_{\mathcal{I}}^2(\chi_i, \chi_j)\right) \end{aligned} \tag{109}$$

with the notations of Section 4.2.1. Thus, for an invertible kernel, one gets

$$\begin{aligned} f\left(d_{\mathcal{I}}^2(\chi_i, \chi_j)\right) &= \frac{1}{2}\left[K_{ii} + K_{jj} - d_F^2(\varphi(\chi_i), \varphi(\chi_j))\right] \\ \Longrightarrow d^2(ij) &= f^{-1}\left[\frac{1}{2}\left\{K_{ii} + K_{jj} - d_F^2(\varphi(\chi_i), \varphi(\chi_j))\right\}\right] \end{aligned} \tag{110}$$

Let $\chi \in \mathcal{I}$ be a test point. Our goal is to find the pre-image $\hat{\chi}$ of the projection $P\varphi(\chi)$ in the kernel PCA space , as shown in Figure 31. Ideally, one has $\varphi(\hat{\chi}) = P\varphi(\chi)$. However, in general, the pre-image may not exist. Hence, one must settle for finding only an approximate pre-image of the projection $P^l\varphi(\chi)$. The reconstructed (approximate) pre-image $\hat{\chi}$ should satisfy the necessary condition of minimizing the distance between the feature point $\varphi(\hat{\chi})$ and the projection in the PCA space $P\varphi(\chi)$, i.e.,

$$\hat{\chi} = \arg\min_{\mathbf{x} \in \mathcal{I}} \| \varphi(\mathbf{x}) - P^l\varphi(\chi) \|^2$$

This can be achieved by minimizing the following error

$$\rho(\hat{\chi}) = \| \varphi(\hat{\chi}) - P^l\varphi(\chi) \|^2$$

The extremum can be obtained by setting $\nabla_{\hat{\chi}}\rho = 0$. In what follows, we derive an expression for reconstructing the pre-image of the projection using different kernels and different types of distances $d_{\mathcal{I}}^2$ in the input space.

4.3.1.4 Computing KPCA Pre-images for the Exponential Kernel

For the exponential kernel in Equation (98), which involves shapes represented as SDFs (denoted by ϕ's), setting $\nabla_{\hat{\phi}}\rho = 0$ gives

$$\hat{\phi} = \frac{\sum_{i=1}^{n} \tilde{\zeta}_i \exp\left(-\| \hat{\phi} - \phi_i \|^2/(2\sigma^2)\right) \phi_i}{\sum_{i=1}^{n} \tilde{\zeta}_i \exp\left(-\| \hat{\phi} - \phi_i \|^2/(2\sigma^2)\right)}, \tag{111}$$

where

$$\zeta_i = \sum_{k=1}^{n} \beta_k u_{ki} \quad \text{and} \quad \tilde{\zeta}_i = \zeta_i + \frac{1}{N}\left(1 - \sum_{j=1}^{N} \zeta_j\right).$$

This is the expression that has been used in the fixed point iteration scheme proposed by Mika *et al.* [80]. As can be easily seen, the pre-image in this case will depend on the starting point and is likely to get stuck in local minima.

To address this issue we make the following approximation

$$\varphi(\hat{\phi}) \approx P^l \varphi(\phi) \tag{112}$$

Further comments on this approximation will be made in the sequel (see, experiments in Section 4.3.1.6). Using the exponential kernel of Equation (98), for which one has $K_{ii} = 1$, in Equation (110), one gets

$$d_{\mathcal{I}}^2(\hat{\phi}, \phi_i) = -2\sigma^2 \log\{\frac{1}{2}(2 - d_F^2(\varphi(\hat{\phi}), \varphi(\phi_i))\}$$

From the relation above, a unique (approximate) pre-image can be computed as

follows

$$\hat{\phi} = \frac{\sum_{i=1}^{n} \tilde{\zeta}_i \exp\left(-d_{\mathcal{I}}^2(\hat{\phi}, \phi_i)/(2\sigma^2)\right) \phi_i}{\sum_{i=1}^{n} \tilde{\zeta}_i \exp\left(-d_{\mathcal{I}}^2(\hat{\phi}, \phi_i)/(2\sigma^2)\right)} = \frac{\sum_{i=1}^{n} \tilde{\zeta}_i \left(\frac{1}{2}(2 - d_F^2(\varphi(\hat{\phi}), \varphi(\phi_i))\right) \phi_i}{\sum_{i=1}^{n} \tilde{\zeta}_i \left(\frac{1}{2}(2 - d_F^2(\varphi(\hat{\phi}), \varphi(\phi_i))\right)}$$
$$\approx \frac{\sum_{i=1}^{n} \tilde{\zeta}_i \left(\frac{1}{2}(2 - d_F^2(P^l\varphi(\phi), \varphi(\phi_i))\right) \phi_i}{\sum_{i=1}^{n} \tilde{\zeta}_i \left(\frac{1}{2}(2 - d_F^2(P^l\varphi(\phi), \varphi(\phi_i))\right)} \tag{113}$$

This equation involves simple algebraic calculations in closed-form (no iteration is required). This is one of main contributions of this work. It should also be noted that, the proposed method is quite different from the one proposed by Kwok [68] which uses multi-dimensional scaling (MDS) and an expensive computation of singular value decomposition (SVD) in the input space to obtain the co-ordinates of the pre-image. Thus, our method not only reduces computation time, but can also be expected to be more accurate since it does not require another level of approximation (projecting back in the input co-ordinate system) to compute the pre-image. Another advantage of the proposed method is that it can be used in almost all scenarios, whereas in the method by Kwok [68], computation of SVD may not be possible if the dimension of the input vector ϕ is large (e.g., for images of very large dimensions).

Using the exponential kernel for shapes represented as binary maps as described in Equation (99) and following the same approach, one gets

$$H\hat{\phi} = \frac{\sum_{i=1}^{n} \tilde{\zeta}_i \left(\frac{1}{2}(2 - d_F^2(P^l\varphi(\phi), \varphi(\phi_i))\right) H\phi_i}{\sum_{i=1}^{n} \tilde{\zeta}_i \left(\frac{1}{2}(2 - d_F^2(P^l\varphi(\phi), \varphi(\phi_i))\right)} \tag{114}$$

Of course, the pre-image appears like a binary map (1 inside, 0 outside) on a 2D grid, and one can find the actual object contour by detecting the points that lie on the 0.5 level set. A different measure of similarity between shapes is given by [46]:

$$d_{\mathcal{I}}^2(\phi_i, \phi_j) = \sum_{p \in Z(\phi_i)} |\phi_j(p)| + \sum_{p \in Z(\phi_j)} |\phi_i(p)|, \tag{115}$$

where $Z(\phi_i)$ is the zero level set of ϕ_i. This distance measure is not a metric on the space of shapes, but allows for partial shape matching. Also note that in this

case, $d_{\mathcal{I}}^2$ is not differentiable and hence a formulation similar to the one derived in the equations above is not possible without assuming a smooth approximation to the absolute value function. However, an empirical formula can be obtained by using the distance measure (115) to calculate distances in input space while using equation (113) for reconstruction.

4.3.1.5 Computing the KPCA Pre-image for the Polynomial Kernel

We now present how a unique (approximate) pre-image may be reconstructed when the polynomial kernel of Equation (100) is used for learning with KPCA. Once again, we set $\nabla_{\hat{\phi}}\rho = 0$, which gives the following expression for obtaining the pre-image of the projection in the KPCA space

$$\hat{\phi} = \sum_{i=1}^{n} \tilde{\zeta}_i \left(\frac{c + \langle \hat{\phi}, \phi_i \rangle}{c + \langle \hat{\phi}, \hat{\phi} \rangle} \right)^{d-1} \phi_i.$$

Using the approximation of Equation (112), which amounts to assuming that

$$d_F^2(P^l\varphi(\phi), \varphi(\phi_i)) \propto d_F^2(\varphi(\hat{\phi}), \varphi(\phi_i)) \ ,$$

the following expression to reconstruct the pre-image is obtained

$$\hat{\phi} = \sum_{i=1}^{n} \tilde{\zeta}_i \left(\frac{\| \ P\varphi(\phi) \ \|^2 + k(\phi_i, \phi_i) - d_F^2(P\varphi(\phi), \phi_i)}{2\| \ P\varphi(\phi) \ \|^2} \right)^{\frac{d-1}{d}} \phi_i \tag{116}$$

where, $\| \ P\varphi(\phi) \ \|^2$ can be easily calculated in terms of the kernel using (see e.g., [68])

$$\| \ P\varphi(\phi) \ \|^2 = \left(\mathbf{k}_x + \frac{1}{N} K\mathbf{1} \right)^T H^T M H \left(\mathbf{k}_x - \frac{1}{N} K\mathbf{1} \right) + \frac{1}{N^2} \mathbf{1}^T K \mathbf{1}.$$

One can note that, if one takes $d = 1$ in the equation above, one gets the expression for performing linear PCA, i.e.,

$$\hat{\phi} = \sum_{i=1}^{n} \tilde{\zeta}_i \ \phi_i \ . \tag{117}$$

Using the methodology described above, one can obtain an expression for computing the pre-image with any invertible kernel and using any type of metric in the input

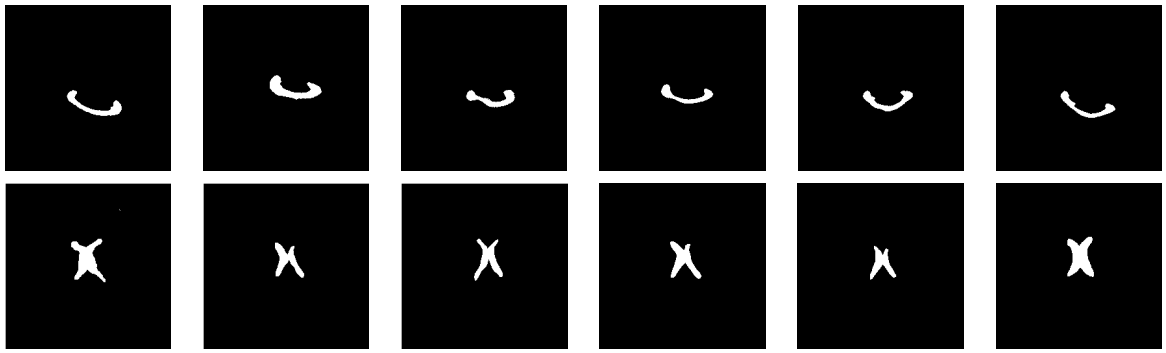
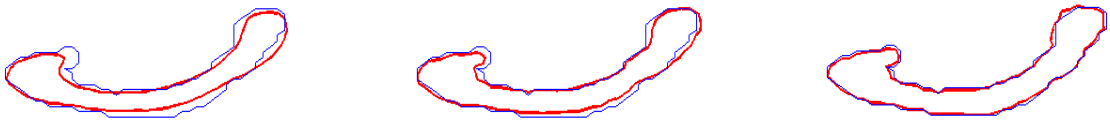


Figure 32: *Top Line:* Samples from the unaligned training set of corpi callosi. *Bottom Line:* Samples from the unaligned training set of ventricles



(a) (b) (c)

Figure 33: Reconstruction of pre-images using different distance functions and representation of shapes: (a)-Euclidean Distance between SDFs; (b)-Euclidean Distance between binary maps and (c)-Distance of Equation (115). Light curve (blue): Original curve. Dark curve (red): Pre-image (result).

space (or representation of shape). In the next section, we perform experiments to evaluate the proposed algorithm. In particular, we use the proposed pre-image approach to perform statistical shape analysis and to provide further experiments confirming the results obtained in Section 4.2.4 above.

4.3.1.6 Pre-Image Experiments

Figure 32 shows some samples from the training set of corpi callosi and of ventricles used in the following experiments. The training set of corpi callosi consists of 23 elements. The training set of ventricles consists of 20 elements. Binary images are shown for better visualization of the variations in shape.

For the experiments in Figure 33, Kernel PCA is performed on the training set

of corpi callosi. The proposed algorithm is used to reconstruct the pre-image of the projection of a test point ϕ. Figure 33 shows the original curve and its pre-image as obtained using an exponential kernel with different distance functions and representations of shapes as mentioned in the preceding section. The reconstruction was performed using $l = 10$. One notes that using the distance measure in Equation (115) leads to slightly better results, on this example. Of course, the choice of σ plays an important role. A typical choice for σ is the average minimum distance between two points in the training set, i.e., $\sigma^2 = \varrho \, \frac{1}{N} \sum_{i=1}^{N} \min_{j \neq i} d_{\mathcal{I}}^2(\phi_i, \phi_j)$, where ϱ is a user defined parameter. As mentioned in Section 4.2.4 above, a larger value of σ allows more "mixing" between elements of the training set, whereas a smaller value of σ uses only a few significant elements in the reconstruction process. This is further demonstrated in Figure 34, where the pre-image is obtained for different values of σ.

Figure 35 shows the first two principal modes of variation of the corpus callosum in the kernel PCA space. It should be noted that this is the first time that such an analysis and visualization has been performed in the context of shapes learnt from applying kernel PCA.

Figure 36 shows the pre-image of the projection of a partially occluded curve. The blue curve is the original data and the red curve is the pre-image of its projection in the KPCA space. An admissible shape is recovered when KPCA is used. The same operation using linear PCA leads to poor results, as shown in the figure. Thus, kernel PCA appears to be quite robust to occlusions and missing information, which can be used advantageously for segmentation or tracking. Next, we further demonstrate the robustness of kernel PCA in learning objects of different geometries (as already seen in Section 4.2.4). The training set of hippocampi and the training set of corpi callosi are combined and the diverse shapes are learned simultaneously. We find the pre-images of the projection $P^l \varphi(\phi)$ of a set of test points.

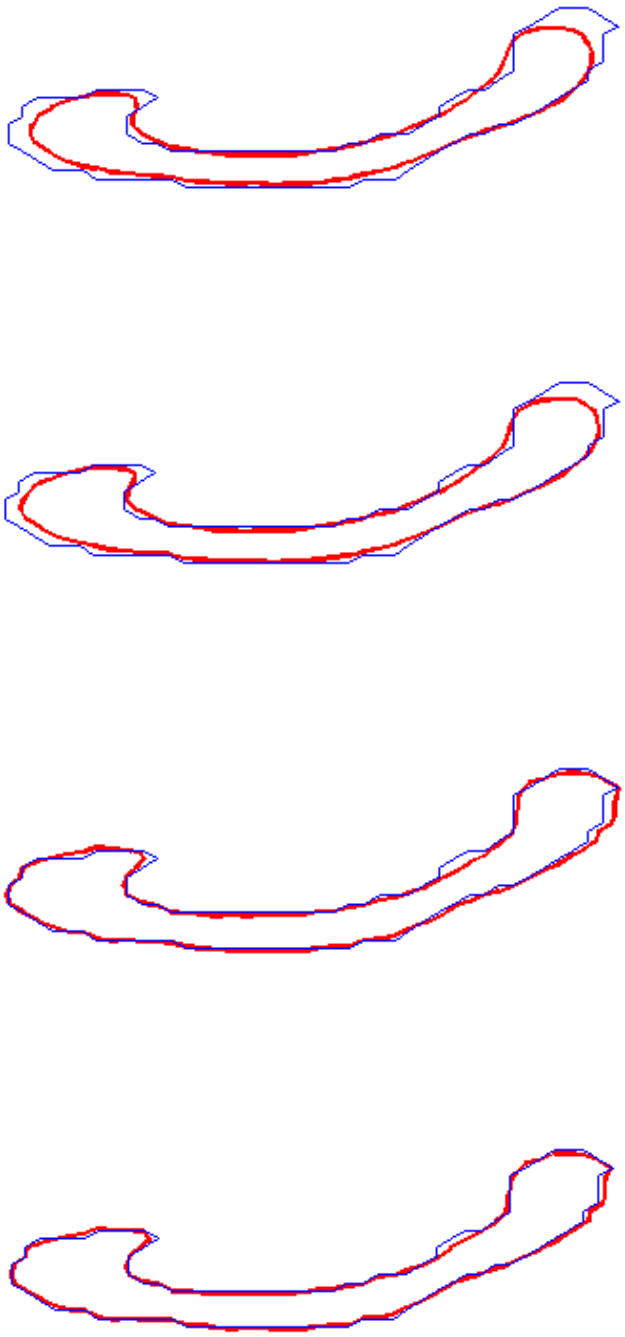
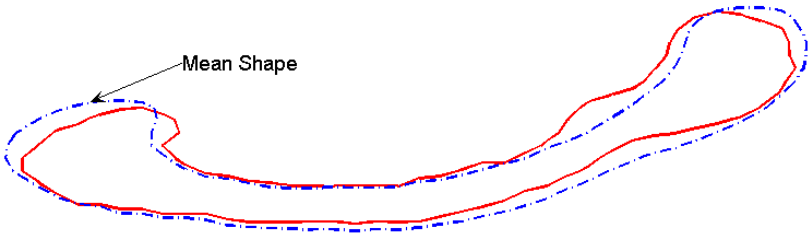
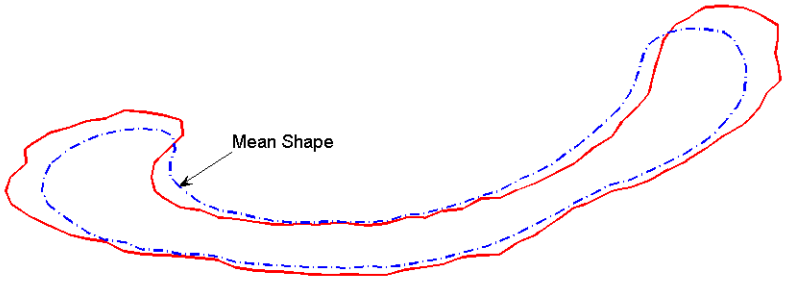


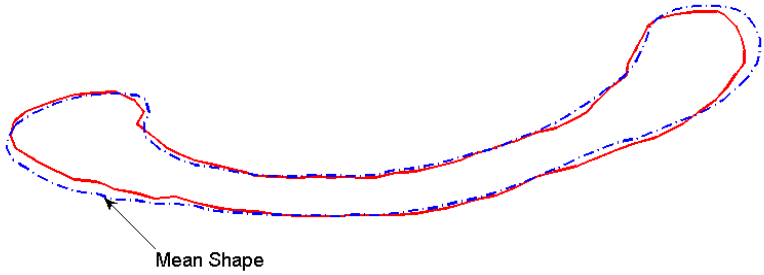
Figure 34: Reconstructed pre-images for different values of σ. (σ increasing from top to bottom). Dark curve (red) is the pre-image result while the lighter curve (blue) is the original.



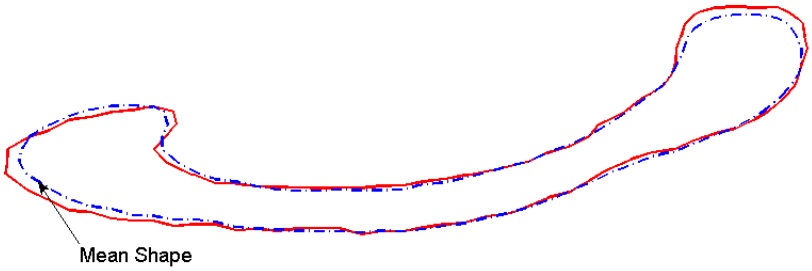
(a) $-2\gamma_1 V_1$ (first mode)



(b) $+2\gamma_1 V_1$ (first mode)

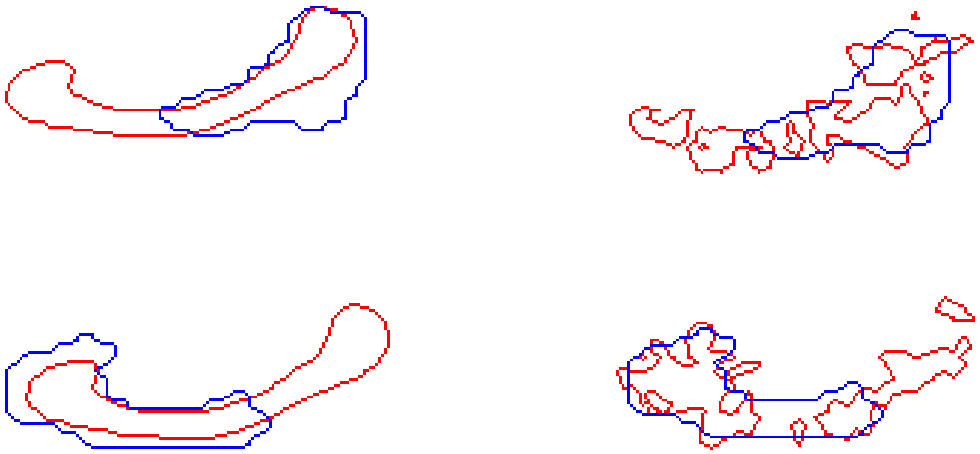


(c) $-2\gamma_2 V_2$ (second mode)



(d) $+2\gamma_2 V_2$ (second mode)

Figure 35: First 2 principal modes of variation of corpus callosum in the feature space. (a) and (b) Shape obtained for $\mp 2\gamma_1 V_1$ variation of the first principal component. (c) and (d) Shape obtained for $\mp 2\gamma_2 V_2$ variation of the second principal component. Dotted curves represent the mean shape.



(a) KPCA (b) Linear PCA

Figure 36: Comparative results in case of missing information. Blue curve: Original (corrupted) curve. Red curve: Result. (a): Pre-image of the projection in the KPCA space; (b): Projection on the linear PCA space.

As can be seen in Figure 37, not only can kernel PCA distinguish between clusters of shapes, it is also robust to partial occlusions (see Figure 37). This property of kernel PCA can be extremely useful in tracking (or segmenting) multiple objects with different shape geometries. Using linear PCA in this experiment would be quite disastrous as is already seen in Figure 36, where only one type of shape is learned. For this experiment, Equation (114) was used.

N.B.: Despite the approximation made in Equation (112), very convincing results are obtained in comparison with linear PCA. Recently the approach proposed in this section was completed in [4]. In [4], the authors use the Nyström extension [7] to better approximate the pre-image obtained with Kernel PCA, for the example of the Gaussian Kernel. The resulting method as well as the method proposed in this section were compared with the method in [67, 80] and linear PCA, notably. Our method and the more exact method in [4] led to satisfying performances in comparison to

(a) ventricles

(b) corpus callosum

Figure 37: Pre-image results when multiple objects are learned simultaneously (corpus callosum *and* ventricles). Light curve (blue): Original curve. Dark curve (red): Pre-image (result). KPCA can distinguish between different shape geometries even if part of the original curve is occluded.

other approaches. Slightly improved performances were observed from the method using Nyström extension, which suggests that the approximation of Equation (112) is reasonable (at least for the diverse experiments performed).

4.3.2 Computing Images for LLE

The Locally Linear Embedding (LLE) algorithm (see [111]) is based on certain simple geometric principles. Suppose the data consists of a training set of N elements $\mathcal{T} = \{\phi_1, ..., \phi_N\}$ sampled from some smooth underlying manifold. Provided there is sufficient data, one can expect each data point and its neighbors to lie on or be close to a locally linear patch of the manifold. One can characterize the local geometry of these patches by a set of coefficients that reconstruct each data point from its

neighbors. In the simplest formulation of LLE, one identifies l nearest neighbors for a data point. Reconstruction error is then measured by the cost function:

$$E(W) = \left(\phi - \sum_j w_j \phi_j \right)^2 \tag{118}$$

where $w = (w_1, ..., w_N)$ is a set of weighting factors. Within the LLE approach, one seeks to minimize the reconstruction error $E(W)$, subject to the constraint that the weights w_j that lie outside the neighborhood are zero and $\sum_j w_j = 1$. With these constraints, the weights for points in the neighborhood of ϕ can be obtained as [35]

$$E(W) = \left(\phi - \sum_{j=1}^{k} w_j \phi_j \right)^2 = \sum_{j=1}^{k} \sum_{m=1}^{k} w_j w_m Q_{jm} \Rightarrow w_j = \frac{\sum_{m=1}^{k} R_{jm}}{\sum_{p=1}^{k} \sum_{q=1}^{k} R_{pq}},$$
$$\text{where } Q_{jm} = (\phi - \phi_j)^T (\phi - \phi_m) \quad \text{and} \quad R = Q^{-1}. \tag{119}$$

In applications where dimensionality reduction is the major objective, one proceeds further and computes a low dimensional vector corresponding to each ϕ_i, preserving the neighborhood structure by keeping the weights w_j constant [35]. This work uses LLE only for obtaining the neighborhood structure in the training set and not for dimensionality reduction. Thus, we assume that a closed surface S can be represented by a linear combination of its k nearest neighbors. Stacking all the columns of ϕ_i one below the other, one can obtain a vector of dimension D^2, if ϕ_i is of dimension $D \times D$. Thus, given a test point ϕ, one can obtain the weights using equation (119) that minimize the reconstruction error $E(W)$. The nearest neighbors are obtained from the training set by finding the squared distance $d_{\mathcal{I}}^2$ in Equation (115) between ϕ and each of the shapes ϕ_i in the training set.

4.3.3 Computing Pre-Images for Kernel LLE

The above LLE algorithm can be generalized for nonlinear manifolds by employing the Mercer kernels [77] and the "kernel trick" [80], leading to the kernel LLE approach. In [31], the author compares the discriminative power of LLE, KLLE and LPCA by

projecting the training data to a lower dimensional space and thereby comparing the recognition rate of a given test sample. The methods presented in this work are quite different than those proposed in [31], since we do not compute a low dimensional data for LLE or KLLE, but compare their performances in the input space itself. This is quite essential for shape analysis in which one needs to compute how accurately a given data point can be reproduced or reconstructed in the input space using these techniques. Thus, the method proposed in [31] uses LLE and KLLE only for classification purposes, while we utilize it to study their performance in the input space of shapes. Another contribution of this work is the formulation of a method to find the pre-image of the projection in the kernel LLE space, given the fact that the mapping φ corresponding to the chosen kernel is unknown.

The basic idea behind KLLE is, given a test point ϕ, to minimize the error

$$E(W) = \left(\varphi(\phi) - \sum_j w_j \varphi(\phi_j) \right)^2 .$$

Proceeding as shown for the case of LLE in the section above, we get the following expression for the weights:

$$w_j = \frac{\sum_{m=1}^{k} R_{jm}}{\sum_{p=1}^{k} \sum_{q=1}^{k} R_{pq}} \quad \text{where,}$$

$$\begin{aligned} Q_{jm} &= (\varphi(\phi) - \varphi(\phi_j))^T (\varphi(\phi) - \varphi(\phi_m)) = k(\phi, \phi) - k(\phi, \phi_m) \\ &\quad - k(\phi, \phi_j) + k(\phi_j, \phi_m) \quad \text{and} \quad R = Q^{-1}. \end{aligned} \tag{120}$$

The weights w_j so obtained minimize the error $E(W)$ in the feature space F.

$$\varphi(\phi) = \underbrace{\sum_{j=1}^{k} w_j \varphi(\phi_j)}_{\Psi} + \sqrt{E} = \Psi + \sqrt{E}.$$

Our goal now is to find the pre-image of Ψ. However, an exact pre-image of Ψ may not exist [80], hence we again settle for an approximate pre-image of Ψ in the input

space $\mathcal{I}$. Thus, we want to find the point $\varphi(\hat{\phi})$ in feature space that is closest to Ψ and for which the pre-image $\hat{\phi}$ can be computed. This can be achieved by minimizing $\rho(z)$, defined as

$$\rho(z) = \parallel \varphi(z) - \Psi \parallel^2 \approx k(z,z) - 2\sum_j w_j k(z,\phi_j) + \langle \Psi, \Psi \rangle$$

Setting $\nabla_z \rho(\hat{\phi}) = 0$ and using the kernel $k(\phi_i, \phi_j) = \exp(-\frac{\|\phi_i - \phi_j\|^2}{2\sigma^2})$, one gets the following expression for finding $\hat{\phi}$:

$$\hat{\phi} = \frac{\sum_{j=1}^{k} w_j k(\hat{\phi}, \phi_j)\ \phi_j}{\sum_{j=1}^{k} w_j k(\hat{\phi}, \phi_j)} \tag{121}$$

This equation contains $\hat{\phi}$ on both sides and hence can be solved by fixed-point iteration technique. However, the solution will depend on the starting point and will be very susceptible to local minima. A unique (but approximate) solution to $\hat{\phi}$ can be found by noting that

$$k(\hat{\phi}, \phi_j) = \langle \varphi(\hat{\phi}), \varphi(\phi_j) \rangle \approx \langle \varphi(\phi), \varphi(\phi_j) \rangle = k(\phi, \phi_j)$$

where we assume $\varphi(\phi) \approx \varphi(\hat{\phi})$. Thus, we have

$$\hat{\phi} = \frac{\sum_{j=1}^{k} w_j k(\phi, \phi_j)\ \phi_j}{\sum_{j=1}^{k} w_j k(\phi, \phi_j)}. \tag{122}$$

Note that this assumption is valid if ϕ is not too dissimilar to the elements of the training set $\mathcal{T}$, which is the case for the out-of-sample experiments we undertook in Section 4.3.4 (The magnitude of the error in the computed pre-image will be function of the error in approximating $\varphi(\hat{\phi}) = \varphi(\phi)$). As shown in [99], better results can be obtained if one uses the distance measure in Equation (115), for $d^2_{\mathcal{I}}$), in the Gaussian kernel instead of the Euclidean L_2 norm. Hence, we use it in all our experiments as described in the next section.

A pre-image can be computed not only for a Gaussian kernel, but for any invertible kernel. If we assume a polynomial kernel $k(\phi_i, \phi_j) = \left(c + \phi_i^T \phi_j\right)^d$, where d is the

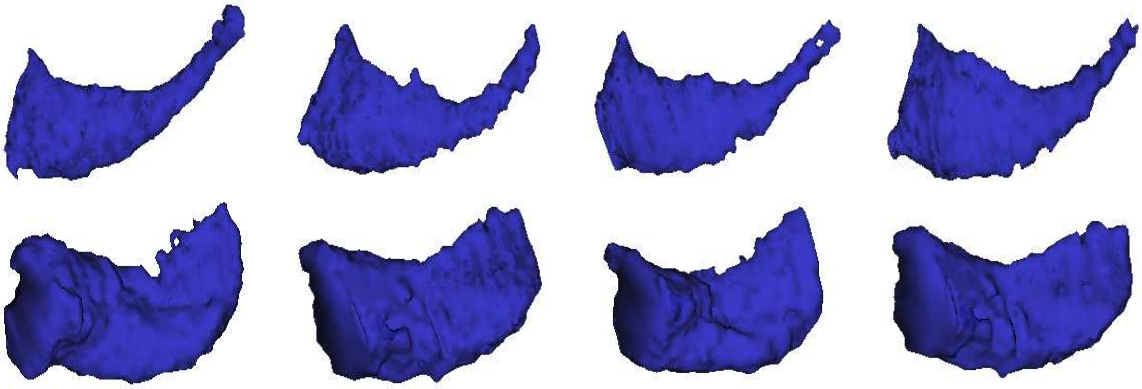


Figure 38: Sample shapes of caudate nuclei (top row) and left hippocampi (bottom row) from the two 3D training sets.

degree of the polynomial and c is any constant, then the pre-image $\hat{\phi}$ of a point $\varphi(\phi)$ is given by

$$\hat{\phi} = \frac{\sum_{j=1}^{k} w_j k(\phi, \phi_j)^{\frac{d-1}{d}} \, \phi_j}{k(\phi, \phi)^{\frac{d-1}{d}}} \tag{123}$$

Thus, LLE is a particular case of KLLE with a polynomial kernel of degree $d = 1$ and $c = 0$. Once again, the k nearest neighbors can be computed using the distance relation (115) or any other metric on the space of shapes [13, 52, 43, 120, 131, 79].

4.3.4 Comparative Analysis

In this section, we describe two experiments to test how well each method (linear PCA, KPCA, LLE and KLLE) performs given a training set of shapes. Two training sets of 3D shapes are used. The first set consists of left caudate nuclei and the second set consists of left hippocampi. These structures in the brain are notoriously challenging to segment and necessitate the use of shape priors in segmentation algorithms. The training set for the caudate nuclei consists of 26 elements, each of them embedded in a signed distance function or a binary volume. The second training set of the hippocampi data contains 22 elements. Figure 38 shows a few shapes from the two training sets.

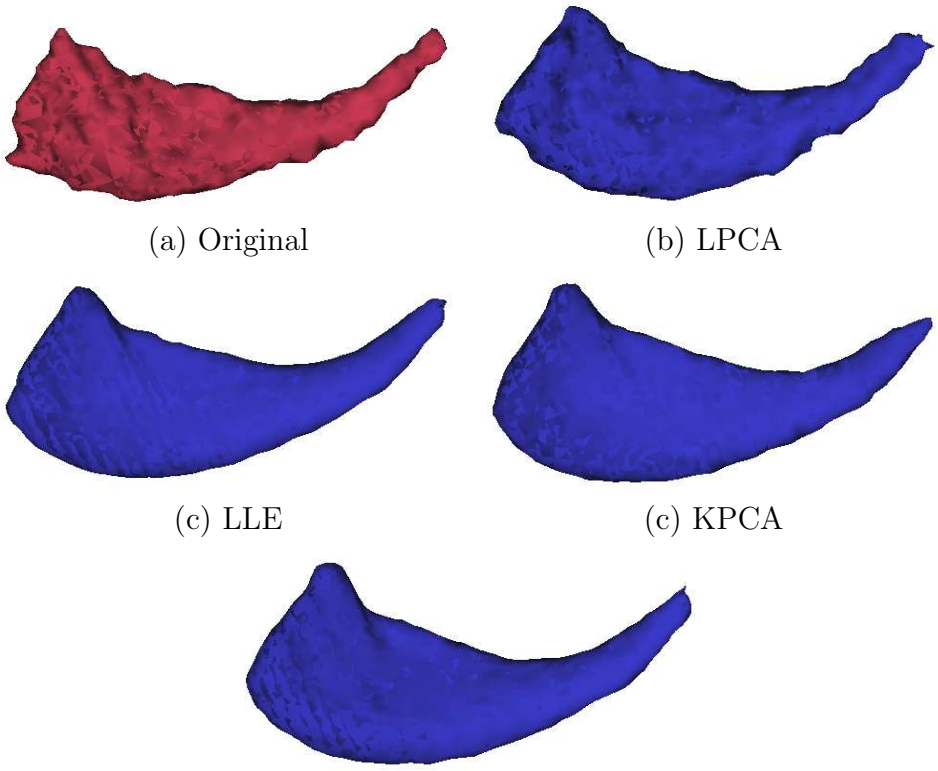
Table 1. mislabeled voxels for left caudate nucleus

Volume	Volume Size	LPCA	LLE	KPCA	KLLE
1	2750	119	50	**37**	42
2	3774	134	105	92	**81**
3	2489	108	66	57	**52**

A typical approach to test the performance of learning methods is to study how well an *unknown* shape , i.e. that is not in the training set, can be represented by each of the methods (out-of-sample problem see e.g., [4]). In this work, a quantitative measure was calculated to compare the performances of the shape learning methods based on linear PCA, KPCA, LLE and KLLE as alluded to above. This measure consists of finding the set symmetric difference between an original test shape and the corresponding pre-image (or projection for linear PCA) obtained from applying each method (this amounts to counting the number of voxels that got mislabeled after reconstruction).

In Figure 39, an unknown shape of a caudate is shown along with the pre-image of the projection obtained using each of the methods. Table 1 shows the number of mislabeled voxels for each of the methods. For LPCA and KPCA, 20 coefficients were used in finding the projection while for LLE and KLLE 20 nearest neighbors were used, as an attempt not to bias results in favor of a particular method. Clearly, the kernel methods perform better than their linear counterparts. More specifically, KLLE performs as well as KPCA, but with a smaller computational burden.

Figure 40 shows the original shape of an hippocampus and the pre-images of projection obtained for each method. For this experiment, we used 15 coefficients for LPCA and KPCA and 15 nearest neighbors for LLE and KLLE. Table 2 gives the number of mislabeled voxels for each of the methods. Figure 41 shows the weights assigned to each of the neighbors (for all the three test shapes) using LLE and KLLE. Clearly, KLLE assigns larger weights to points closer to the test shape than to points farther away. Thus, only points in the locally linear patch of the feature space are

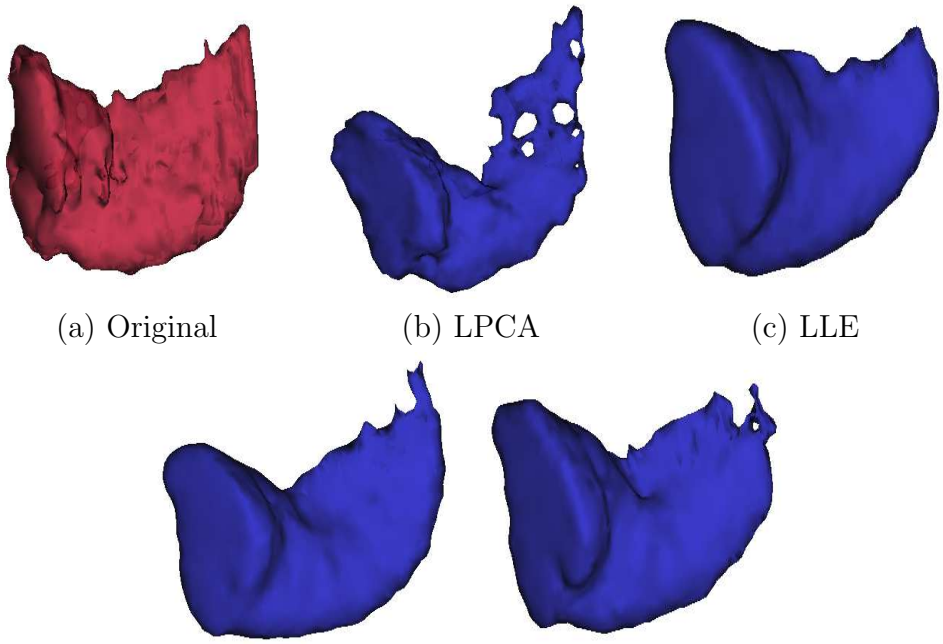


(a) Original (b) LPCA

(c) LLE (c) KPCA

(e) KLLE

Figure 39: Projection of left Caudate nucleus (Volume 1) using each of the methods.



(a) Original (b) LPCA (c) LLE

(d) KPCA (e) KLLE

Figure 40: Projection of left Hippocampus (Volume 3) using each of the methods.

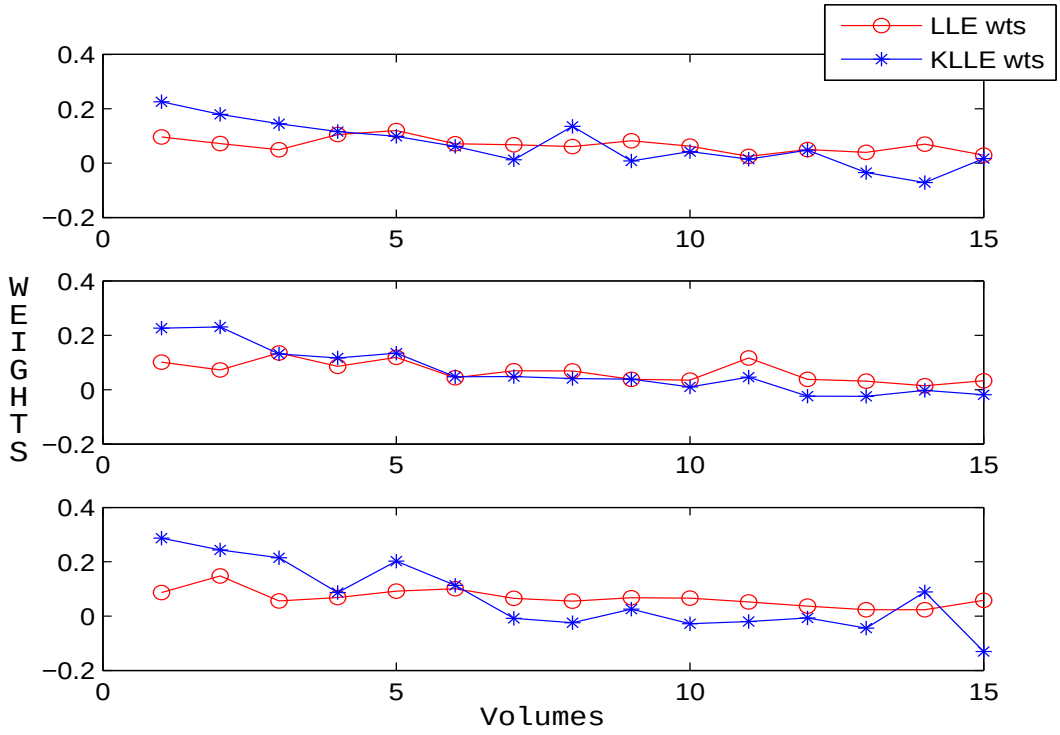


Figure 41: Weights assigned to the 15 nearest neighbors by LLE and KLLE for each of the test shapes of hippocampus. On the x-axis, 1 is the closest neighbor, 15 is the farthest.

assigned significant weights, whereas other points are assigned weights close to zero. This nonlinear distribution is expected since we used a Gaussian kernel. Once again, it is clear that KLLE and KPCA perform better than all the linear methods. It should be noted that, LLE and KLLE can perform even better with the proper choice of the number of nearest neighbors as given in [35]. To make a fair assessment of each method, we kept k (nearest neighbors) fixed and did not optimize the algorithm as given in [35].

Table 2. mislabeled voxels for left hippocampus

Volume	Volume Size	LPCA	LLE	KPCA	KLLE
1	1117	440	378	322	**296**
2	1108	306	258	212	**205**
3	1568	804	574	494	**371**

In all of the experiments above, the parameter σ used in the Gaussian kernel was fixed to be some function of the average minimum distance between shapes in the training set [21], i.e., $\sigma^2 = c\ \frac{1}{n}\sum_{i=1}^{n} \min_{j \neq i} d_{\mathcal{I}}^2(\phi_i, \phi_j)$., where c is a user defined real number. The training data (hand segmented shapes) was obtained from the NAMIC data repository of the Brigham and Women's Hospital, Boston, MA. The entire code was written in C++ using the ITK and VTK libraries.

4.3.5 Remarks

In this section, we have proposed new algorithms for finding an approximate pre-image of the projection of a point in the kernel space in the context of Kernel LLE and KPCA, notably. The resulting approaches were compared in terms of their capability to represent unseen shapes, notably. Experiments show that kernel approaches performs better than LPCA and LLE. Approaches based upon KPCA and KLLE led to comparable performance (but the KLLE approach necessitates fewer computations).

Representing a shape using its nearest neighbors, as is done in LLE and KLLE approaches, requires that the training set contains a sufficient number of data points. LPCA and KPCA have an intrinsic capability to "produce" shapes by varying the PCA coefficients. This is not the case with LLE or KLLE. On the other hand, if sufficient amount of data is available, LLE and KLLE may perform as well or better than PCA based algorithms. Another advantage of LLE, KLLE and KPCA is that they allow one to learn shapes of completely different geometries, within the same training set. The reason for this is that these methods use only their nearest neighbors to find the projection instead of using the entire training set as is the case with LPCA.

4.4 Chapter Conclusion

In this chapter, we proposed to use diverse machine learning techniques (Linear PCA, Kernel PCA, LLE and Kernel LLE) to introduce shape priors in the GAC framework. Two main avenues were explored: a direct approach that realizes optimization directly in feature space and an indirect approach that computes pre-images of projections in the learned space of shape. The advantage of the indirect approach is that it is computationally efficient since shape-models can be computed in closed form. However, approximations need to be made to compute pre-images. On the other hand, the direct approach involves a more costly (iterative) optimization process, but it can be expected to be more accurate since it does not necessitate any approximation.

In both direct and indirect approaches, the projections on the learned spaces of shapes play a crucial role as shape-models. Using the projections on the learned spaces indeed appear to be a sensitive choice when dealing with multi-modal distributions, for instance (in contrast with fitting a statistical model to the shape space that can bias shape priors towards unrealistic shapes). Further, using such models obtained from projections allows for comparing learning techniques in a rigorous manner.

The performances of the diverse learning methods were compared in terms of accuracy, robustness and ability to learn and recognize different objects of various shapes. The superiority of nonlinear (kernel based) learning techniques over linear ones was highlighted by numerous examples in multiple situations. Hence, nonlinear techniques should always be preferred to their linear counterparts in practical application (their computational cost being moreover comparable). Furthermore, the choice whether to adopt the direct or indirect approaches should be made based on the application: if speed is essential (e.g., military applications), the indirect method can be preferred; if precision is essential (e.g., medical applications) the direct method should be adopted.

Among the learning techniques studied, the Kernel PCA based approach stands out as a valuable method for introducing shape priors in the geometric active contour framework. In particular, the method was shown to be able to elegantly deal with very complex variations in shapes, to simultaneously learn multiple objects and to constrain the shape of contours in a meaningful, accurate and robust manner.

The focus of future work could be to investigate the performance of other kernels or to study the influence of other shape metrics to introduce shape priors within the geometric active contour framework. Furthermore, devising automated methodologies to choose optimally among different kernels to perform learning would also be an interesting avenue of research.

CHAPTER V

COMBINING NONLINEAR SHAPE PRIOR AND INTENSITY INFORMATION

In this chapter, we combine methods presented in Chapters 3 and 4, to perform image segmentation with nonlinear shape priors, within the geometric active contour (GAC) framework. This chapter is based on parts of [28, 26] and is organized as follows: In the next section, we provide some motivation and related work to the approach presented in this chapter. In Section 5.2, we detail and motivate our approach to combine image cues and nonlinear shape information in a meaningful fashion. In Section 5.3, we present an efficient method that performs the necessary step of aligning the shapes in the training set with the object in the image. Finally, in Section 5.4, we provide experimental segmentation results of challenging artificial images and real-world sequences.

5.1 Motivation and Related Work

In this chapter, we combine image and shape information to perform the segmentation of challenging images. The use of shape prior in addition to extracting image information is oftentimes necessary when parts of the object of interest are missing or when the statistics (or local features such as edges) of the object are undistinguishable from the ones of the background. Figure 42, presents two such images: In the first one, the person (soccer player) is hardly distinguishable from the background using pixels statistics or edges. In the second one, the word "yellow" is present but some of the letters are missing and occluded (see also Section 4.1, for further motivation

concerning the use of shape priors).

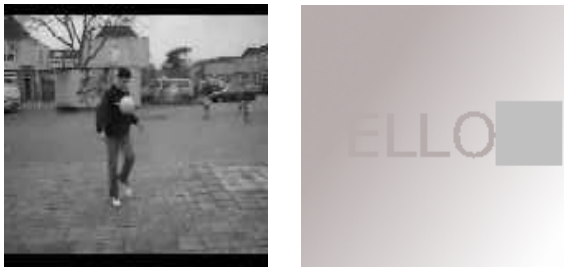


Figure 42: Two examples of images where the use of shape priors is necessary in addition to image information. *Left :*Image information only is not enough to distinguish the object (soccer player) from the background. *Right :*Image information is missing (letters of the word "yellow" are occluded).

Two main classes of approaches to combine image cues and shape knowledge can be distinguished, within the active contour methodology. In the first class, one combines two energy functionals (each of which respectively encodes image information and shape knowledge) and evolves a complete (parametric or implicit) contour (see for instance, [16, 70, 107, 20, 21, 22, 25, 14]). Within the first class, interesting recent advancements propose to locally enforce the shape prior in regions of the image [23] or more or less strongly (at the pixel level) based on some confidence measure [108, 121]. In the second class, one performs optimization in the finite dimensional space of learnt shapes and evolves only a few coordinate vectors to perform (constrained) segmentation (see for example, [124, 105, 85, 84, 102]). Within the second class, algorithms present the advantage of being quite computationally efficient. However, if training sets are too small or not diverse enough to be representative of all the possible shapes of the object, the first class of approaches should be preferred since it is less dependent on the quality of the training sets.

In this chapter, we opted for an approach belonging to the first class. The proposed framework contrasts with previously proposed approaches in many aspects. Indeed, we exploit the nonlinear shape learning method proposed in Chapter 4 to introduce shape priors into the GAC framework. This enables the segmenting curve to undergo

topological changes in a seamless manner. Also, and contrary to methods in [22, 70] for instance, no assumption is made on the distribution (e.g., Gaussian) in feature space of the diverse shapes, which can have the advantage of not biasing the shape of the contour towards the mean shape notably. In addition, we exploit the shape interpretation of the approach presented in Chapter 3 that uses image information to separate regions. In particular, this allows us to balance image cues and shape knowledge in a meaningful fashion, a widely overlooked point in the literature. Indeed, within our methodology *both* image and shape information are described in terms of shapes. This consistent description allows us to combine energies to realize meaningful tradeoffs. Furthermore, a very efficient registration scheme arises from the shape interpretation of our image energy that enables us to consider only the transformation of the image instead of requiring to transform all the shapes in the training sets (as in most previously proposed approaches).

5.2 Balancing energy functionals

In this section, we combine shape knowledge and image information using the methodologies presented in the two preceding chapters. In the remainder, E^F_{shape} denotes the energy functional presented in Equation (102) using the kernel of Equation (99) that allows one to perform nonlinear PCA on binary maps (see Chapter 4). Also, E_{image} denotes the energy functional in Equation (29), in which G is the result of thresholding the image. Contrary to Chapter 3, we do not take the variation of the functional G with respect to the segmenting curve C into account, i.e. we take $\mathcal{V}_{\text{geom}} = 0$. This amounts to assuming that pixel statistics inside and outside the curve vary slowly with respect to C. This is done to highlight the ability of E_{shape} *only* to constrain the shape of the contour. We refer to the functional G as the "image-shape model", in what follows, to insist on the fact that G is indeed a shape extracted from the image. The Heaviside and Dirac delta functions are mollified as described in Equation (97)

of Chapter 4.

As alluded to in the section above, image and shape information can be combined by defining a total Energy E_{total}:

$$E_{\text{total}}(I, \phi) = E^{\text{F}}_{\text{shape}}(\phi) + \beta . E_{\text{image}}(I, \phi) \tag{124}$$

Using Bayes' formula, it can be shown that minimizing the energy functional E_{total} of Equation 124 amounts to maximizing the posterior probability of the curve given current image information and shape knowledge. Another interpretation of the minimization of E_{total} is that of a constrained minimization of E_{image}, subject to the constraint that E^{F}_{shape} is minimal, via the Lagrange multiplier $\frac{1}{\beta}$. The coefficient β represents the level of trust given to image information (higher β's result in emphasizing image information). Considering $-\nabla_{\phi} E_{\text{total}}$ in Equation (124), two forces are obtained: One is the image force $f_{\text{image}} = -\beta \nabla_{\phi} E_{\text{image}}$, the other is the shape force $f^{\text{F}}_{\text{shape}} = -\nabla_{\phi} E^{\text{F}}_{\text{shape}}$. The gradient descent process described in Equation (12) converges to a local minimum where f_{image} and $f^{\text{F}}_{\text{shape}}$ balance each other (same amplitude and opposite directions).

However, E_{image} is a squared L_2-distance in input space, whereas $E^{\text{F}}_{\text{shape}}$ is a squared distance in feature space. As a consequence, a meaningful balance between f_{image} and $f^{\text{F}}_{\text{shape}}$ would be hard to reach. The gradient $\nabla_{\phi} E^{\text{F}}_{\text{shape}}$ exhibits a nonlinear behavior due to the exponential terms it involves. To illustrate this point, one can consider a 1-dimensional static version of E_{image} and $E^{\text{F}}_{\text{shape}}$.[1] Let $E^{\text{1d}}_{\text{image}} = (x - x_{\text{image}})^2$ be the 1-dimensional analogous of E_{image}, with x_{image} the 1-dimensional equivalent to the image-shape model and let $E^{\text{1d}}_{\text{shape}} = 2 - 2e^{\frac{(x - x_{\text{shape}})^2}{2\sigma^2}}$ be the 1-dimensional analogous of $E^{\text{F}}_{\text{shape}}$, with x_{shape} the 1-dimensional equivalent to the shape model obtained from

[1] This discussion involving 1-dimensional functions also holds for multiple dimensions since the different forces enjoy radial symmetry.

learning[2]. Although this is usually not the case during the actual evolution process, the models x_{image} and x_{shape} will be considered constant throughout this discussion, for simplicity. The 1-dimensional forces $f^{\text{1d}}_{\text{image}}$ and $f^{\text{1d}}_{\text{shape}}$ corresponding respectively to $E^{\text{1d}}_{\text{image}}$ and $E^{\text{1d}}_{\text{shape}}$ can easily be computed as

$$f^{\text{1d}}_{\text{image}} = -2\beta.(x - x_{\text{image}}) \quad \text{and} \quad f^{\text{1d}}_{\text{shape}} = -\frac{2}{\sigma^2}(x - x_{\text{shape}})e^{\frac{(x-x_{\text{shape}})^2}{2\sigma^2}}.$$

The amplitudes and directions of $f^{\text{1d}}_{\text{image}}$ and $f^{\text{1d}}_{\text{shape}}$ are presented in Figure 5.2, for fixed values of x_{image}, x_{shape} and σ as well as four different values of β. The amplitude of $f^{\text{1d}}_{\text{shape}}$ presents two maxima around x_{shape}. The values of x, denoted x_{max1} and x_{max2}, for which these maxima occur are: $x_{\text{max1}} = x_{\text{shape}} - \sigma$ and $x_{\text{max2}} = x_{\text{shape}} + \sigma$. During gradient descent, the variable x evolves, from an initial value of $x = x_0$, in the direction of the force that has the highest amplitude to converge to the value of $x = x_{\text{final}}$ for which $f^{\text{1d}}_{\text{image}}$ and $f^{\text{1d}}_{\text{shape}}$ have same amplitudes and opposite directions. Although the graphs presented in Figure 5.2 are dependent on the particular choice of x_{image}, x_{shape} and β, valuable insights about the consequences of the nonlinear behavior of $E^{\text{F}}_{\text{shape}}$ may be inferred:

- If β is too large, $|f^{\text{1d}}_{\text{image}}|$ (■-line on the graph) is always above $|f^{\text{1d}}_{\text{shape}}|$: the evolution of x is not influenced by $f^{\text{1d}}_{\text{shape}}$, and x should converge to x_{image}. Hence, it can be expected that for too large β's, no shape-knowledge will contribute at all to the contour evolution, and the contour will converge to the image-shape model. This is counter-intuitive, since from Equation (124), one expects that *both* energies contribute to the evolution (for $\beta \neq 0$).

- If the initial value of x is biased towards x_{image}, for instance x_0 located on the right of x_{image}, $f^{\text{1d}}_{\text{shape}}$ will not contribute to the evolution of x, and convergence

[2]Recall that $E^{\text{F}}_{\text{shape}}$ is a squared distance d^2_{F} in the feature space. We assume here for the purpose of illustration that $P^l\varphi(x)$ has a pre-image x_{shape} in the input space, i.e., $\varphi(x_{\text{shape}}) = P^l\varphi(x)$. Thus, the 1-dimensional energy $E^{\text{1d}}_{\text{shape}} = 2 - 2k(x, x_{\text{shape}}) = 2 - 2e^{\frac{(x-x_{\text{shape}})^2}{2\sigma^2}}$. This assumption is not true in general; see further remarks in what follows.

again occurs for $x = x_{\text{image}}$. Thus, it can be expected that the final contour will be very much dependent on initial conditions. For example, if the shape of the initial contour is biased towards the image-shape model the contour could converge towards the image-shape model and shape-knowledge information would not influence the evolution.

Hence, the darker area under the $|f^{\text{1d}}_{\text{shape}}|$-curve represents the locus of the possible final values of x realizing a tradeoff between image information and the knowledge of shape. These possible final values are between x_{shape} and x_{max2}. Final results for x will thus be skewed towards the shape-knowledge model, and it is not possible to reach any tradeoff value located between x_{max2} and x_{image} (for any initial value x_0). Thus, the parameter σ can be expected to have a limiting influence on the possible tradeoffs realizable between image information and shape-knowledge, resulting in not meaningfully balanced final contours. Ideally, *only* the parameter β should influence the tradeoff between forces, while σ should be chosen based on considerations of the training data *only*.

Figure 5.2 presents the typical behavior of the L_2-norm of $f^{\text{F}}_{\text{shape}}$, obtained when warping an initial contour very "far from" (i.e., dissimilar to) the elements of the training set, until convergence. This empirical result for the norm of $f^{\text{F}}_{\text{shape}}$ exhibits very similar behavior as its 1-dimensional theoretical counterpart $|f^{\text{1d}}_{\text{shape}}|$, validating the remarks made above[3].

Hence, these issues need to be addressed to take advantage of the kernel PCA technique, in a robust (e.g.: less dependant on initial conditions), intuitive and meaningful manner. We now proceed to explain a methodology to address the aforementioned

[3]Also, during our initial experiments using Equation (124) directly, meaningful balances between shape knowledge and image information were difficult to reach (despite investigating wide ranges of β). Final contours were either biased towards the image-shape model or adopted a familiar shape that did not segment the image in a satisfactory manner. Satisfactory balances were obtained only for large values of σ.

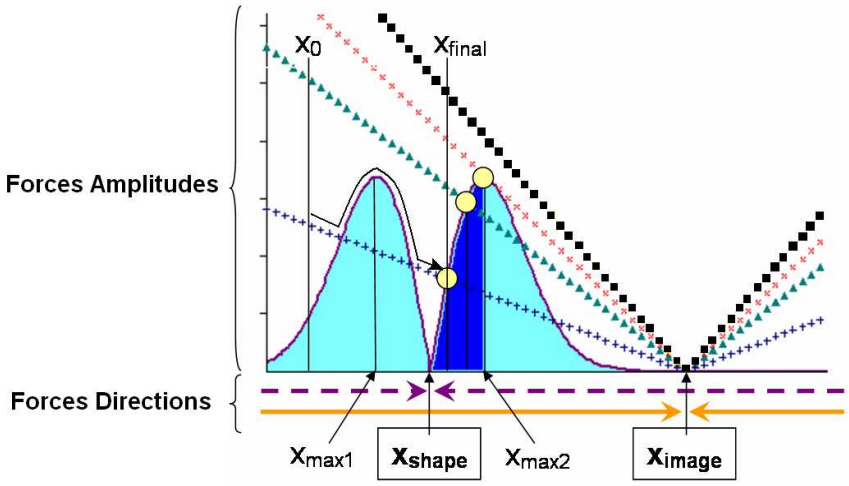


Figure 43: Theoretical forces corresponding to E_{image} and $E^{\text{F}}_{\text{shape}}$. The double-bell-shaped curve represents $|f^{\text{1d}}_{\text{shape}}|$. The two-segments curves represent $|f^{\text{1d}}_{\text{image}}|$ for diverse values of β (higher absolute slopes correspond to higher β's). The respective directions of both forces are materialized by the two pairs of opposite horizontal arrow at the bottom of the figure. The darker area under the $|f^{\text{1d}}_{\text{shape}}|$-curve represents the locus of possible convergence points that realize a tradeoff between $|f^{\text{1d}}_{\text{image}}|$ and $|f^{\text{1d}}_{\text{shape}}|$: For the $+, \times$ and $\triangle$-lines these corresponding possible point of convergence are indicated by a circle (No trade-off is realizable between $|f^{\text{1d}}_{\text{image}}|$ and $|f^{\text{1d}}_{\text{shape}}|$, for the $\blacksquare$-line)

issues, for the exponential kernel (The methodology is general enough to accommodate any invertible kernel of the form $k(x, y) = k(\|x - y\|)$). The nonlinear behavior of $\nabla E^{\text{F}}_{\text{shape}}$ stems from the fact that the energy is defined as a distance in feature space. We would like to redefine the shape energy so that the amplitude of its gradient exhibits a linear behavior (similarly to ∇E_{image} for which the energy is defined as a distance in input space). We note that distances in the feature space and the input space are related as follows, for any SDFs ϕ_a and ϕ_b:

$$d^2_F(\phi_a, \phi_b) = 2 - 2k_{\varphi_\sigma}(H\phi_a, H\phi_a) = 2 - 2e^{-\frac{\|H\phi_a - H\phi_b\|^2}{2\sigma^2}} = 2 - 2e^{-\frac{d^2_I(H\phi_a, H\phi_b)}{2\sigma^2}} \quad (125)$$

where d^2_I is a squared distance in the input space. Using the invertibility of the kernel in Equation (125), one can write

$$d^2_I(H\phi_a, H\phi_b) = -2\sigma^2 \log(\frac{2 - d^2_F(\phi_a, \phi_b)}{2}). \quad (126)$$

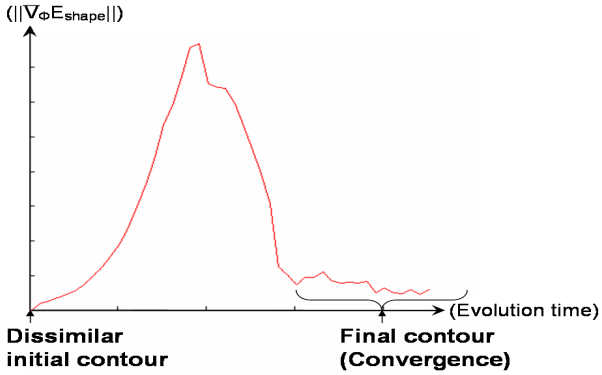


Figure 44: Experimental result highlighting the nonlinear behavior of the L_2 norm of $\nabla_\phi E^{\mathrm{F}}_{\mathrm{shape}}$. An initial contour "far away" from the learned shapes was warped (note the similarity between the experimental curve and the theoretical solid-line curve in the leftmost graph).

Let us define a new shape energy from $E^{\mathrm{F}}_{\mathrm{shape}}$:

$$E_{\mathrm{shape}} = -2\sigma^2 \log\left(\frac{2 - E^{\mathrm{F}}_{\mathrm{shape}}}{2}\right). \tag{127}$$

This energy can be expected to behave in a similar fashion as a squared distance in input space. The gradient of the energy E_{shape} should now exhibit a more stable behavior than $\nabla E^{\mathrm{F}}_{\mathrm{shape}}$. By applying the chain rule, one can verify that

$$\nabla_\phi E_{\mathrm{shape}} = \frac{2\sigma^2}{2 - E^{\mathrm{F}}_{\mathrm{shape}}} \nabla_\phi E^{\mathrm{F}}_{\mathrm{shape}}.$$

Hence, $\nabla E^{\mathrm{F}}_{\mathrm{shape}}$ and $\nabla E_{\mathrm{shape}}$ have identical direction ($\frac{2\sigma^2}{2-E^{\mathrm{F}}_{\mathrm{shape}}} \in \mathbf{R}^+$, since $E_{\mathrm{shape}} \in [0, 2]$) and will influence the evolution in similar ways since both gradients point in the direction of the projection in feature space. However, the extra term in the expression of $\nabla_\phi E_{\mathrm{shape}}$, namely $\frac{2\sigma^2}{2-E^{\mathrm{F}}_{\mathrm{shape}}}$, modifies the amplitude of the gradient and was found to indeed drastically improve balancing performances between image and shape forces (for a rather large range of values of σ), during our experiments.

It is important to note that for a certain ϕ, $P^l\varphi(\phi)$ does not necessarily have a pre-image in the input space $\mathcal{I}$: It may not be possible to define $\phi_P \in \mathcal{I}$, so that $\phi_P = \varphi^{-1}(P^l\varphi(\phi))$. Thus, rigorously defining an energy functional as a distance between the contour and a model encoding shape knowledge obtained from kernel

PCA directly in the input space is problematic. A few authors proposed methods to compute approximate pre-image of $P^l\varphi(\phi)$. In [67], an iterative scheme is proposed to compute $\phi_P^{\text{approx}} \simeq \varphi^{-1}(P^l\varphi(\phi))$, while in [99], a direct method is proposed. These methods could have been used to define, $E_{\text{shape}}^{\text{approx}} = \|H\phi - H\phi_P^{\text{approx}}\|^2$, directly in $\mathcal{I}$. However, defining E_{shape}, as the scaled log of a distance in the feature space, was found to be more stable while not requiring any approximation of the shape model.

Both the shape and image energies now behave as distances in input space between shapes that are represented by binary maps. This consistent description of energies allows for meaningful tradeoffs between image cues and shape knowledge, through the energy functional

$$E_{\text{total}}(\phi, I) = (1-\alpha)\, E_{\text{shape}}(\phi) + \alpha\, E_{\text{image}}(\phi, I), \tag{128}$$

where α is a balancing parameter, effectively and intuitively encoding the level of trust accorded to each model (image-shape model or shape-knowledge model). For example, if the same level of trust should to be accorded to each model during segmentation, α can be chosen to be .5.

5.3 Pose Invariance

Prior to the construction of the space of shapes, the elements of the training set need to be aligned, e.g., to disregard differences due to affine transformations. As a result, the elements of the training set are registered at a certain position in the image domain (usually at the center of the domain). Typically, the object of interest in the image I and the registered training shapes do not have the same pose and differ by some transformation. This transformation needs to be recovered during the segmentation process to constrain the shape of the contour properly (see e.g., [70, 124]). In what follows, we use similar notations as in [124], and detail our approach to deal with differences in pose within our segmentation framework.

Let us assume that the object of interest in I differs from the registered elements of the training set by the transformation $T[\mathbf{p}]$ with parameters $\mathbf{p} = [p_1, p_2, p_3, p_4] = [t_x, t_y, \theta, \rho]$, in which t_x and t_y correspond to translation according to x and y-axis, θ is the rotation angle and ρ the scale parameter. Let us denote by $\tilde{I}(\tilde{x}, \tilde{y})$ the image obtained by applying the given transformation on $I(x, y)$. We have

$$\tilde{I}(\tilde{x}, \tilde{y}) = I(x, y)$$

with

$$(\tilde{x}, \tilde{y}) = T[\mathbf{p}](x, y) = (\rho(x\cos\theta - y\sin\theta) + t_x, \rho(x\sin\theta + y\cos\theta) + t_y)$$

The transformation $T[\mathbf{p}]$ may be recovered by minimizing $E_{\text{total}}(\phi, \tilde{I})$ with respect to the p_i's, via gradient descent. Since $E_{\text{shape}}(\phi)$ does not depend on $\mathbf{p}$, minimizing $E_{\text{total}}(\phi, \tilde{I})$ is equivalent to minimizing $E_{\text{image}}(\phi, \tilde{I})$ only, with respect to the p_i's. In particular, we have

$$\nabla_{p_i} E_{\text{total}}(\phi, \tilde{I}) = \nabla_{p_i} E_{\text{image}}(\phi, \tilde{I})$$

for $i = 1, \ldots, 4$. Assuming that the image-shape model G is negligibly modified by an infinitesimal transformation of $\tilde{I}$[4], one can compute $\nabla_{p_i} E_{\text{image}}(\phi, \tilde{I})$ by considering that $E_{\text{image}}(\phi, \tilde{I})$ is a distance between two determined binary maps, namely $H\phi(x, y)$ and $G(\tilde{x}, \tilde{y}) = G(T[\mathbf{p}](x, y))$. One can thus write

$$\nabla_{p_i} E_{\text{image}}(\phi, \tilde{I}) = 2\int_{\Omega} (H\phi - G)\nabla_{p_i} G dx dy, \tag{129}$$

where

$$\nabla_{p_i} G = \Big[\frac{\partial G(\tilde{x}, \tilde{y})}{\partial \tilde{x}} \quad \frac{\partial G(\tilde{x}, \tilde{y})}{\partial \tilde{y}}\Big]\Big(\frac{\partial T[\mathbf{p}](x, y)}{\partial p_i}\Big)^t.$$

To perform segmentation of images with shape priors, the following gradient descent scheme is undertaken in parallel to the contour evolution of Equation (12), for

[4]This amounts to assuming that the pixel statistics inside and outside the curve (computed for $\tilde{I}$) are constant for an infinitesimal transformation of $\tilde{I}$.

$i = 1, \ldots, 4$:

$$\frac{dp_i}{dt} = -\nabla_{p_i} E_{\text{image}}(\phi, \tilde{I}). \tag{130}$$

To initiate the segmentation process, an estimate $\mathbf{p}_0$ of $\mathbf{p}$ needs to be computed (the initial curve is drawn around the object in $\tilde{I}$), so that the pose of the object of interest in $\tilde{I}$ roughly matches the pose of one of the registered training shapes. Recovering the transformation $T[\mathbf{p}]$ by working with the image I *only* instead of transforming each element of the training set is different with previously proposed approaches (e.g., [21, 124]) and is computationally very advantageous when dealing with large training sets.

5.4 Experiments, segmentation with nonlinear shape prior

This section presents segmentation results obtained by introducing shape prior using KPCA on binary maps and using our region-based segmentation methodology. For all the following experiments, Equation (12) was run until convergence on diverse images, using Equation (128) as an energy functional (the parameter α was chosen to be $\alpha = 50\%$). In parallel, the transformation $T[\mathbf{p}]$ between the object of interest and the registered training set was recovered. No curvature term was used for contour regularization since smoothness information is indeed already present in the binary maps used to build shape priors. The training sets used in the experiments are the "4 words", "Soccer Player" and "Shark" training sets as presented in Section 4.2.4.1 of Chapter 4.

5.4.1 Toy example: shape priors involving objects of different types

The goal of this section is to investigate the ability of the algorithm to accurately *recognize* and segment objects of different shapes. For these experiments the "4 Words" training set was used. We tested our algorithm on images where a corrupted version of either of the four words "orange," "yellow," "square" or "circle" was present

(Figure 45, first row). Word recognition is a challenging task and addressing it using geometric active contours may not be a panacea. However, the ability of the level set representation to naturally handle topological changes was found to be useful for this purpose. During evolution, the contour split and merged a certain number of times to segment the disconnected letters of the words. In all the following experiments, the *same* initial contour was used.

Experiment 1: In this experiment, the word "square" was corrupted: The letter "u" was almost completely erased. The shape thus obtained was filled with gaussian noise of mean $\mu_o = .5$ and variance $\sigma_o = .05$. The background was also filled with Gaussian noise of same mean $\mu_b = .5$ but of variance $\sigma_b = .2$. The result of applying our method is presented Figure 45(a). The image-shape model was built using the method involving two (constrained) thresholds as described in Section 3.4.1.2. Despite the noise and the partial deletion, a convincing segmentation and smooth contour are obtained. In particular, the *correct font* is detected and the letter "u" accurately reconstructed.

Experiment 2: In this second experiment, one of the elements of the training set was used. A thick line (occlusion) was drawn on the word and a fair amount of gaussian noise was added to the resulting image. The result of applying our method is presented Figure 45(b). Despite the noise and the occlusion, a very convincing segmentation is obtained. In particular, the *correct font* is detected and the thick line completely removed. Once again, the final contour is smooth despite the fairly large amount of noise.

Experiment 3: In this third test, the word "yellow" was written using a *different* font from the ones used to build the training set. In addition, "linear shadowing" was used in the background, making the first letter "y" completely hidden. The letter "w" was also replaced by a grey square. The result of applying our framework is

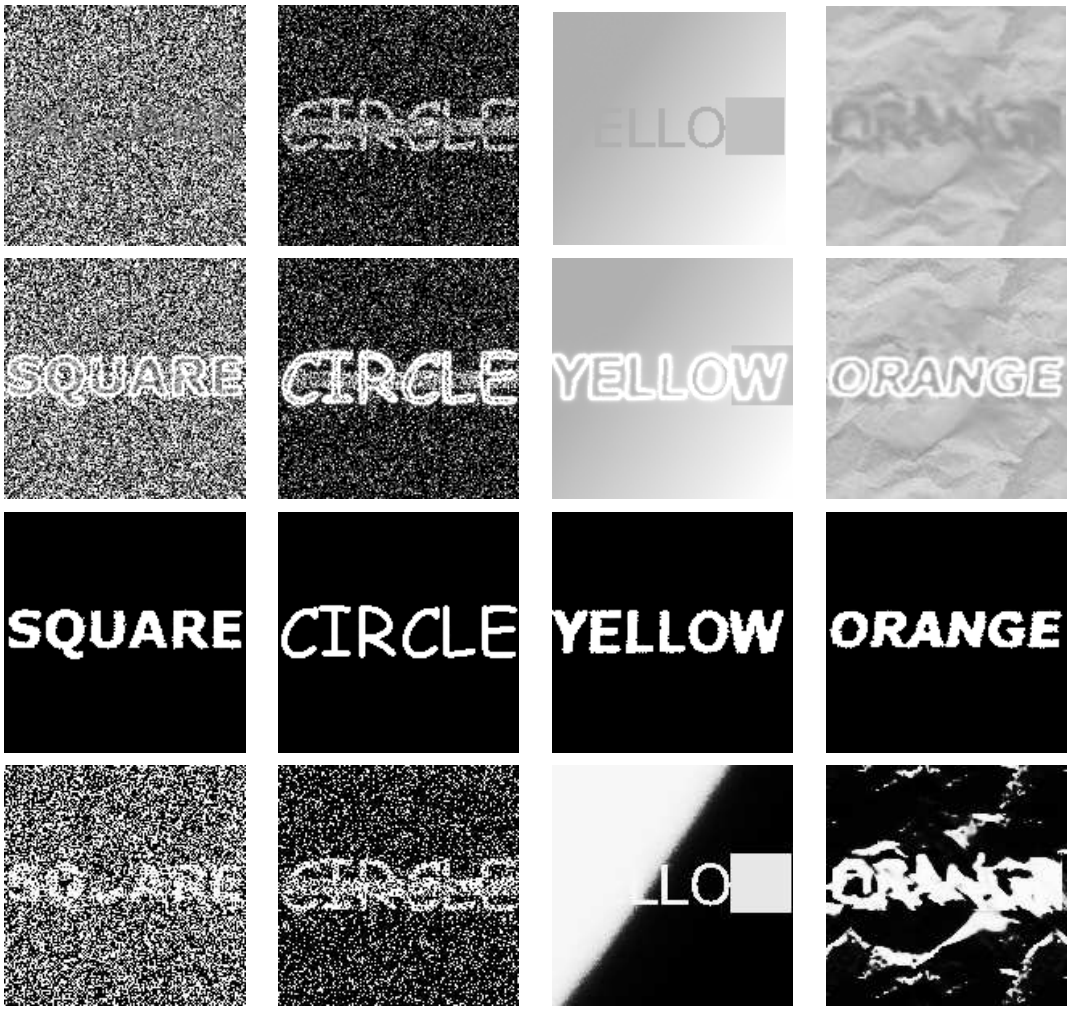
presented in Figure 45(c). The word "yellow" is correctly recognized and segmented. In particular, the letters "y" and "w", were completely reconstructed.

Experiment 4: In this experiment, the word "orange" was *handwritten* in capital letters roughly matching the size of the letters of the words in the training set. The intensity of the letters was chosen to be rather close to some parts of the background. In addition, the word was blurred and smeared in a way that made its letters barely recognizable. This type of blurring effect is often observed in medical images due to patient motion. This image is particularly difficult to segment, even using shape prior, since the spacing between letters and the letters themselves are very irregular due to the combined effects of handwriting and blurring. Hence, mixing among classes (confusion between either of the 4 words) can be expected in the final result. In the final result obtained, the word "orange" is not only recognized but satisfyingly recovered; in particular, a thick font was obtained to model the thick letters of the word (Figure 45(d)).

Hence, starting from the same initial contour in each experiment, our algorithm was able to accurately detect which word was present in the image. This highlights the ability of our method not only to gather image information throughout evolution but also to distinguish between objects of different classes ("orange," "yellow," "square," and "circle"). Comparing the final contours obtained in each experiments to the final image-shape model $G_{[I,\phi(t)]}$ (last row of Figure 45), one can measure the effect of our shape prior model in constraining the contour evolution. The image information alone would lead to a shape bearing very little resemblance to any of the learned words.

5.4.2 Real images examples: tracking temporal sequences

To test the robustness of the framework, tracking was performed on two sequences. A very simple tracking scheme was used with the *same* initial contour was used for each image in the sequence. This contour was initially positioned wherever the final contour



(a) (b) (c) (d)

Figure 45: Segmentation results for the "4 Words" training set. Shape Priors were built by applying the kernel PCA on binary maps. First row: Original images to segment, Second row: Segmentation results, Third row: shape underlined by the final contour ($H\phi$), Fourth row: image-shape model (G) obtained when computing E_{image} for the final contour.

was in the preceding image. Of course, many efficient tracking algorithms have already been proposed. However, convincing results were obtained here, without considering the system dynamics, for instance. This highlights the efficiency of including prior knowledge on shape for the robust tracking of deformable objects. The parameter α was *fixed* to $\alpha = 55\%$ in Equation (128) to make up for the influence of clutter in the images. The nice tracking performance observed, despite the fact that parameters are fixed throughout the sequence, further highlight the robustness of the proposed segmentation framework.

5.4.2.1 Soccer player sequence

In this sequence (composed of 130 images), a man is playing with a soccer ball. The challenge is to accurately capture the large deformations due to the movement of the person (e.g., limbs undergo large changes in aspect), while sufficiently constraining the contour to discard clutter in the background. The training set of 22 silhouettes, presented on the first row of Figure 23, was used. The version of E_{image} involving the intensity means only was used to capture image information. Despite the small number of shapes used for training and the initialization of the contour evolution with the same (arbitrary) contour, successful tracking was obtained, correctly capturing the very diverse postures of the player.

5.4.2.2 Shark video

In this sequence (composed of 70 images), a shark is evolving in a highly cluttered environment. Note that the shark is at times partially occluded by other fish and has poorly contrast. To perform tracking, 15 shapes were extracted from the first half of the video (Figure 23, second row) and used to build shape prior. The version of E_{image} involving global statistics was used to make up for the poor contrast of the shark in the images. Once again, despite the small training set, successful tracking

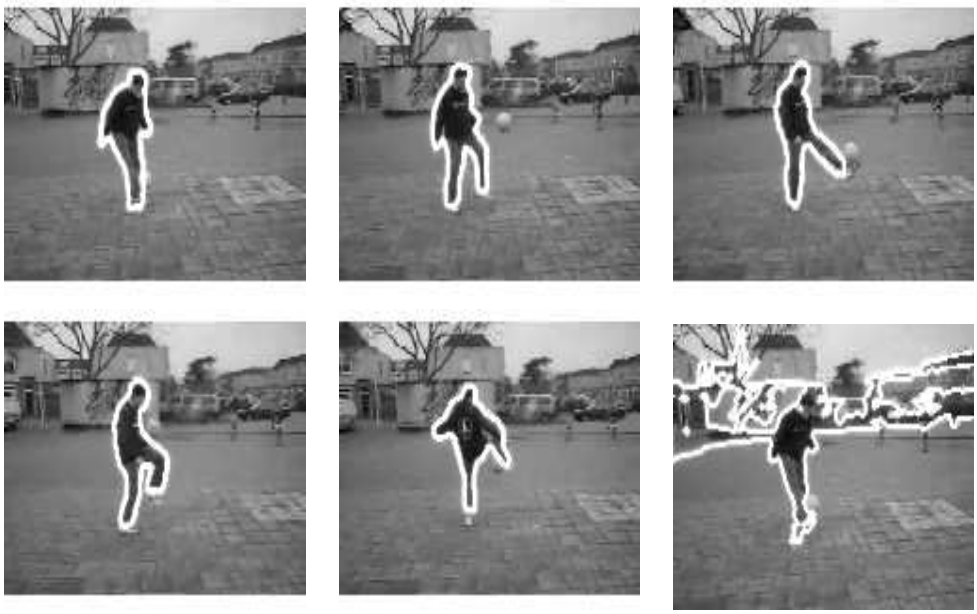


Figure 46: Tracking results with the proposed method for the Soccer Player Sequence (Bottom rightmost frame is the result obtained without shape prior; $\alpha = 1$ in (128)).

performances were observed. The shark was correctly captured, while clutter and occlusions were rejected.

5.5 Chapter Conclusion

In this chapter, we used an unsupervised and nonlinear learning technique to introduce shape priors in the geometric active contours framework and guide segmentation. Our approach employs the projections on the learned spaces of shapes, obtained using kernel PCA, as models. In addition, we used a region-based approach aiming at separating an object from the background by extracting an image-shape model from the image.

Furthermore, we highlighted the importance of expressing image and shape energies in a consistent manner. This is especially important when dealing with nonlinear techniques such as kernel PCA. Valuable insights about the pitfalls of balancing forces extracted directly from the feature space with "linear" (image) forces were obtained,

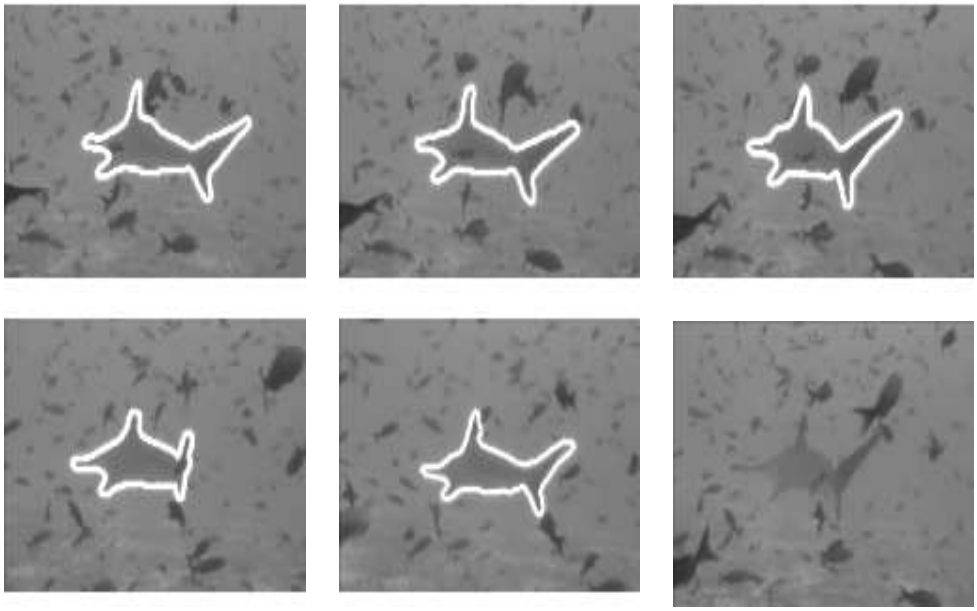


Figure 47: Tracking results with the proposed method for the "Shark" sequence (Bottom rightmost frame is an original image from the sequence, reproduced here to assess the poor level of contrast).

from the study of 1-dimensional approximations. A method for addressing some of the potential problems was also proposed, which drastically improved balancing performances. In the proposed algorithm, statistical information extracted from images and shape prior knowledge are combined in a consistent fashion, since both energies can be understood in terms of shapes. This allows one to realize meaningful trade-offs and fuse heterogenous sources of information in an efficient manner.

Very reasonable segmentation performances were obtained on complex images, highlighting the power and accuracy of the proposed framework. For instance, the method was shown to be robust to noise, blurring, occlusion and clutter. Moreover, the proposed framework was proven to be able to recognize and choose *without supervision* the appropriate models (among the shapes corresponding to different objects learned simultaneously) that should be used to perform the segmentation of images featuring a corrupted version of the object of interest.

CHAPTER VI

2D REGION-BASED SEGMENTATION AND 3D POSE ESTIMATION FROM A 3D SHAPE PRIOR

In this chapter, we address the problem of jointly segmenting a 3D object of arbitrary shape in a 2D image and estimating its 3D pose with respect to a referential attached to a unique calibrated camera. This chapter is organized as follows: In the first section, we provide some motivation and related work to the proposed approach. In the second section, we describe our variational methodology and detail the main aspects of the proposed flow. In the third section, we present experimental results that highlight the robustness of the technique to initialization, noise, occlusion and clutter, with a special emphasis on challenging tracking applications. Finally, in the fourth section, we discuss some properties of the proposed approach, draw our conclusions and present possible extensions.

6.1 Motivation and Related Work

2D image segmentation and 2D-3D pose estimation are ubiquitous tasks in computer vision applications and have received a great deal of attention in the past few years. In this work, we combine both approaches in a variational framework. To appreciate the contribution of this work, we recall some of the results and specifics of both fields.

2D-3D pose estimation aims at determining the pose of a 3D object relative to a calibrated camera from one unique or a collection of 2D images. Knowing the mapping between the world coordinates and image coordinates from the camera calibration matrix, and after establishing correspondences between 2D features of the object

in the image and their 3D counterparts on the model, it is possible to solve the pose transformation from a set of equations that express these correspondences. The literature concerned with 3D pose estimation is very large and a complete survey is beyond the scope of this chapter. Most methods can be distinguished by the type of *local* image features used to establish correspondences, such as points [97], lines or segments [32, 75], multi-part curve segments [137], or complete contours [104, 36].

Segmentation consists of separating an object from the background in a given image. The geometric active contour (GAC) framework, in which a curve is evolved continuously to capture the boundaries of an object, was proven to be quite successful at performing this task. Originally, the method focused on extracting local image features such as edges to perform segmentation; see [10, 63] and the references therein. However, edge-based techniques can suffer from the typical drawbacks that arise from using local image features: high sensitivity to noise or missing information, and a multitude of local minima that result in poor segmentations. Region-based approaches, which use global image statistics inside and outside the contour, were shown to drastically improve the robustness of segmentation results [139, 12, 93, 29]. Region-based techniques are able to deal with various statistics of the object and background such as distinct mean intensities [12], Gaussian distributions [93, 29] or intensity histograms [78, 64] as well as a wide variety of photometric descriptors such as grayscale values, color or texture [91]. Further improvement of the GAC approach consists of learning the shape of the object to be segmented and constrain the contour evolution to adopt a familiar shape, to make up for poor segmentation results obtained in the presence of noise, clutter, occlusion or when the statistics of the object and background are difficult to distinguish (see e.g., [70, 124, 21, 26]).

Motivation/Contribution: Our goal is to combine the strengths of both techniques and to avoid some of their typical weaknesses, to robustly both segment 2D images and estimate the pose of an arbitrary 3D object which shape is known.

In particular, we use a region-based approach to continuously drive the pose estimation process. This global approach avoids using local image features and, hence, addresses two shortcomings that typically arise from doing so in most 2D-3D pose estimation algorithms: Firstly, finding the correspondence between local features in the image and on the model is usually non-trivial, due for instance to their viewpoint dependency - no local correspondences need to be found in our global approach. Secondly, local image features may not even exist or can be difficult to detect in a reliable and robust fashion in the presence of noise clutter or occlusion. Furthermore, more or less strong assumptions usually need to be made on the class of shapes that a 2D-3D pose estimation technique can handle. Many approaches are limited to relatively simple shapes that can be described using geometric primitives such as corners, lines, circles or cylinders. Recent work focused on free-form objects which admit a manageable parametric description as in [104]. However, even this type of algebraic approaches can become unmanageable for objects of arbitrary and complex shape. Our approach can deal with object of *arbitrary* shape, represented by a 3D level set [87] or a 3D cloud of points (see Figure 49).

Conversely, a shortcoming of the GAC framework using shape priors is that 2D shapes are usually learned to segment 2D images. Hence, a large collection of 2D shapes needs to be learned to represent the wide variation in aspect that most natural 3D objects take when projected onto the 2D image plane. Our region-based approach benefits from the knowledge of the object shape that is compactly described by a *unique* 3D model. In addition, and in contrast to the GAC framework, the proposed method does not involve the evolution of an infinite dimensional contour to perform segmentation, but only solves for the finite dimensional pose parameters (as is common for 2D-3D pose estimation approaches). This results in a much simplified framework that avoids dealing with problems such as infinite dimensional curve representation, evolution and regularization .

Relation to Previous Work: In this chapter, we exploit many ideas from recent variational approaches that address the problem of structure from motion and stereo-reconstruction from multiple cameras ([133, 134] or [44]). Originally, the authors in [133, 134] presented a method to reconstruct the 3D shape of an object from multiple 2D views obtained from calibrated cameras: the relative 3D pose of the cameras could also be recovered. The present contribution aims at performing a somewhat opposite task: given the 3D model of an object, perform the segmentation of 2D images and recover the 3D pose of the object relative to a *unique* camera. This is the first time that the framework in [133, 134] is adapted and employed in the specific context of segmenting 2D images from a unique camera, using the knowledge of a 3D model. The framework in [134] has also recently been extended in [125] to address the problem of multiple camera calibration. In the present work, the camera is assumed to be calibrated. However, this assumption could easily be dropped by also solving for the optimal camera calibration parameters as presented in [125].

We note that, although the use of 3D shape knowledge to perform the 2D segmentation of regions presents obvious advantages, the literature dealing with this type of approaches is strikingly thin. The closest piece of work to the proposed contribution is probably [103]. In [103], the authors evolve an (infinite dimensional) active contour as well as 3D pose parameters to minimize a joint energy functional encoding both image information and 3D shape knowledge. Our method differs from the aforementioned approach in many crucial aspects. For example, we optimize a *unique* energy functional, which allows us to circumvent the need to determine ICP-like correspondences and to perform costly back-projections between the segmenting contour and the shape model at each iteration. Also, we perform optimization *only* in the finite dimensional space of the Euclidean pose parameters. In addition to being computationally efficient, this allows our technique to be less likely to be trapped in local minima, resulting in robust performances as demonstrated in the experimental part.

Furthermore, our approach relies on surface differential geometry (see e.g., [33]) to link geometric properties of the model surface and its projection in the image domain. This allows us to derive the partial differential equations necessary to perform energy optimization.

Our technique uses a 3D shape prior in a region-based framework, and can thereby be expected to be robust to noise or occlusion. Hence, an obvious application of the proposed approach is the robust tracking of 3D rigid objects in 2D image sequences. Our approach is, thus, also related to a wealth of methods concerned with the problem of model-based monocular tracking (see [69] for a recent survey).

6.2 Proposed Approach

We suppose we have at our disposal the 3D surface model of an object. Our goal is to find the (Euclidean) transformation that needs to be applied to the model so that it coincides with the object of interest in the referential attached to a calibrated camera. We now describe the proposed approach in details, starting with our choice of notation. An overview of the method can be found in Figure 48.

6.2.1 Notation

Let $\mathbf{X} = [X, Y, Z]^T$ denote the coordinates of a point in $\mathbb{R}^3$, measured with respect to a referential attached to the imaging camera. We denote by I the image, by $\Omega \subset \mathbb{R}^2$ the image domain, and by $d\Omega$ its area element. We assume the camera is modeled as an ideal perspective projection[1]: $\pi : \mathbb{R}^3 \mapsto \Omega; \mathbf{X} \mapsto \mathbf{x}$, where $\mathbf{x} = [x, y]^T = [X/Z,\ Y/Z]^T$ denotes coordinates in Ω.

Let S be the smooth surface in $\mathbb{R}^3$ defining the shape of the object of interest. The (outward) unit normal to S at each point $\mathbf{X} \in S$ will be denoted by $\mathbf{N} =$

[1]More general models of cameras (see [45, 56]) can straightforwardly be handled. We make this assumption here to simplify the presentation.

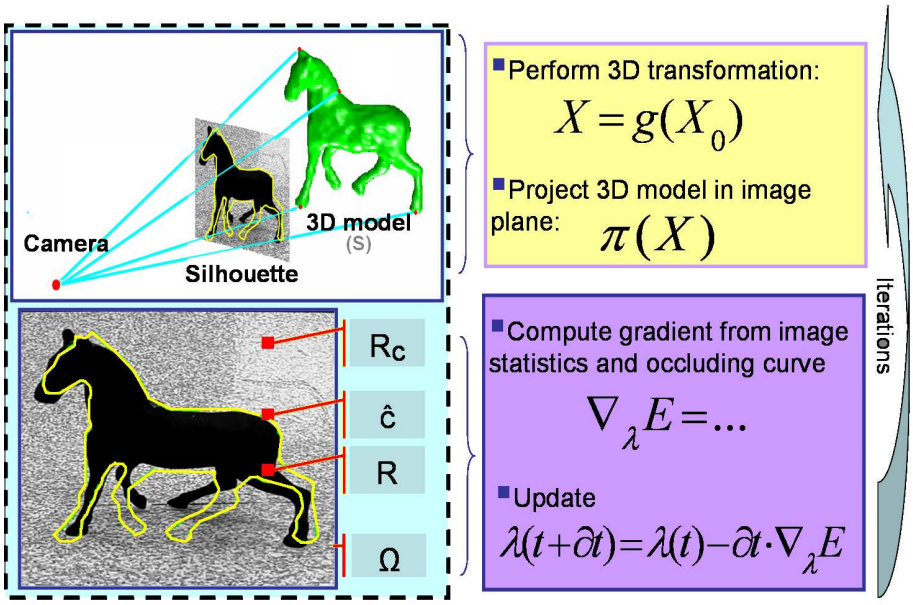


Figure 48: Schema summarizing our segmentation/pose estimation approach from a 3D model, in 4 steps.

$[N_1, N_2, N_3]^T$. To determine the pose of S with respect to the camera, we define the identical reference surface S_0, whose pose is known.[2] Denoting by X_0 the coordinates of points on S_0, one can locate S in the camera referential via the transformation $g \in SE(3)$, such that $S = g(S_0)$, or written point-wise

$$\mathbf{X} = g(\mathbf{X}_0) = \mathbf{R}\mathbf{X}_0 + \mathbf{T}$$

with $\mathbf{R} \in SO(3)$ and $\mathbf{T} \in \mathbb{R}^3$. The parameters of the rigid motion g will be denoted by $\lambda = [\lambda_1, \ldots, \lambda_6]^T = [t_x, t_y, t_z, \omega_1, \omega_2, \omega_3]^T$ (where rotations are represented using exponential coordinates, see [73]).

Let $R = \pi(S) \subset \Omega$, be the region of the image on which the surface S projects (i.e., the region of Ω corresponding to imaging S). Let $R^c = \Omega \setminus R$ and $\hat{c} = \partial R$ denote the complement and the boundary of R, respectively (Figure 48). The curve $\hat{c} \subset \Omega$ is the projection of the curve $C \subset S$ that delineates the visible part of S from the

[2] One can assume that the center of gravity of S_0 coincides with the camera center and the rotation is known.

camera: $\hat{c} = \pi(C)$. The 2D curve $\hat{c}$ and 3D curve C will be respectively referred to as the "silhouette" and the "occluding curve".

6.2.2 Energy functional

In [134], the authors used an image formation approach to define a cost functional measuring the discrepancy between the photometric properties of the surface S (as well as the 3D-background), and the pixel intensities of multiple images. The resulting energy involved back-projections to the surface S to guarantee the coherence between the measurements obtained from multiple cameras. In this present work, we are interested in segmenting a *unique* image and adopt a somewhat different approach directly inspired from region-based active contours techniques [12, 93, 29, 78]. Most region-based approaches assume that the pixels corresponding to the object of interest or the background are distinct with respect to a certain grouping criterion. To perform segmentation, a closed curve is evolved to increase the discrepancy between the statistics of the pixels located in its interior and exterior. Accordingly, we define a energy of the form:

$$E = \int_{\mathrm{R}} r_O\big(I(\mathbf{x}), \hat{c}\big) d\Omega + \int_{\mathrm{R^c}} r_B\big(I(\mathbf{x}), \hat{c}\big) d\Omega, \tag{131}$$

where $r_O : \mathcal{Z}, \Omega \mapsto \mathbb{R}$ and $r_B : \mathcal{Z}, \Omega \mapsto \mathbb{R}$ are two monotonically decreasing functions measuring the matching quality of the image pixels with a statistical model over the regions R and R^c, respectively. The space $\mathcal{Z}$ corresponds to the photometric variable (such as gray-scale intensity, color or texture vector) chosen to perform segmentation. Hence, depending on the choices for r_O, r_B, and $\mathcal{Z}$ a larger class of images than the ones fitting the specific hypotheses made in [134] can be dealt with. The energy E measures the discrepancy between the statistical properties of the pixels located inside and outside the curve $\hat{c}$, and does not involve any back-projections.

It is expected that E is minimal when R and R^c correspond to the object and background in I, respectively. Most region-based approaches evolve an infinite dimensional curve, which amounts to exploring unconstrained shapes of the segmenting contour. Since we assume that the 3D shape of the rigid object is known, we want to minimize E by exploring only the possible regions R and R^c that result from projecting the surface S onto the image plane. For a calibrated camera, these regions are functions of the transformation g only. Solving for the transformation that minimize E can be undertaken via gradient descent over the parameters λ, as described below.

6.2.3 Gradient flow

The partial differentials of E with respect to the pose parameters λ_i's can be computed using the chain-rule:

$$\frac{dE}{d\lambda_i} = \int_{\hat{c}} \Big(r_O\big(I(\mathbf{x})\big) - r_B\big(I(\mathbf{x})\big) \Big) \left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} + \int_R \left\langle \frac{\partial r_O}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega + \int_{R^c} \left\langle \frac{\partial r_B}{\partial \hat{c}}, \frac{\partial \hat{c}}{\partial \lambda_i} \right\rangle d\Omega, \tag{132}$$

where $\hat{s}$ is the arc-length parametrization of the silhouette $\hat{c}$ and $\hat{\mathbf{n}}$ the (outward) normal to the curve at $\mathbf{x}$.

For the first term in (132), using the arc-length s of C and the direct $\frac{\pi}{2}$-rotation matrix J, one has

$$\begin{aligned} = \left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} &= \left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, J\frac{\partial \hat{c}}{\partial \hat{s}} \right\rangle d\hat{s} \\ &= \left\langle \frac{\partial \pi(C)}{\partial \lambda_i}, J\frac{\partial \pi(C)}{\partial s} \right\rangle \frac{ds}{d\hat{s}} d\hat{s} = \left\langle \frac{\partial \pi(C)}{\partial \lambda_i}, J\frac{\partial \pi(C)}{\partial s} \right\rangle ds. \end{aligned} \tag{133}$$

Let $\mathcal{J}$ denote the Jacobian of $\pi(\mathbf{X})$ with respect to the spatial coordinates, one has

$$\mathcal{J} = \frac{1}{Z^2} \begin{bmatrix} Z & 0 & -X \\ 0 & Z & -Y \end{bmatrix}$$

From (133), one gets

$$\left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} = \left\langle \mathcal{J}\frac{\partial \mathbf{X}}{\partial \lambda_i}, J\mathcal{J}\frac{\partial \mathbf{X}}{\partial s} \right\rangle ds = \left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathcal{J}^T J \mathcal{J}\frac{\partial \mathbf{X}}{\partial s} \right\rangle ds$$
$$= \frac{1}{Z^3}\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \begin{bmatrix} 0 & Z & -Y \\ -Z & 0 & X \\ Y & -X & 0 \end{bmatrix} \frac{\partial \mathbf{X}}{\partial s} \right\rangle ds = \frac{1}{Z^3}\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \frac{\partial \mathbf{X}}{\partial s} \times \mathbf{X} \right\rangle ds. \tag{134}$$

In the equation above, the point $\mathbf{X}$ belongs to the occluding curve C. A necessary condition for a point $\mathbf{X}$ to belong to the occluding curve is that $\langle \mathbf{X}, \mathbf{N} \rangle = 0$ (since the associated vector $\mathbf{X}$, with origin at the center of the camera, corresponds to the projection/viewing direction and is tangent to the surface S at $\mathbf{X}$). The vector $\mathbf{t} = \frac{\partial \mathbf{X}}{\partial s}$ is the tangent to the curve C at the point $\mathbf{X}$. Since the vectors $\mathbf{t}$ and $\mathbf{X}$ belong to the tangent plane to S at $\mathbf{X}$, one has

$$\frac{\partial \mathbf{X}}{\partial s} \times \mathbf{X} = \|\mathbf{X}\| \mathbf{N} \sin(\theta)$$

with $\theta = (\widehat{\mathbf{t}, \mathbf{X}})$ the angle between $\mathbf{t}$ and $\mathbf{X}$. For $\mathbf{X} \in C$, one has

$$\begin{aligned} \frac{\partial}{\partial s}\langle \mathbf{X}, \mathbf{N} \rangle &= \underbrace{\left\langle \frac{\partial \mathbf{X}}{\partial s}, \mathbf{N} \right\rangle}_{=0} + \left\langle \frac{\partial \mathbf{N}}{\partial s}, \mathbf{X} \right\rangle = 0 \\ &= \langle d\mathbf{N}(\mathbf{t}), \mathbf{X} \rangle = \mathrm{II}(\mathbf{t}, \mathbf{X}). \end{aligned} \tag{135}$$

Hence, since the second fundamental form $\mathrm{II}(\mathbf{t}, \mathbf{X}) = 0$, the vectors $\mathbf{t}$ and $\mathbf{X}$ are conjugate (see [33]). Hence, using the Euler formula, one can show that

$$K \sin^2 \theta = \kappa_X \kappa_t$$

where K is the Gaussian curvature, and κ_X and κ_t denote the normal curvatures in the directions $\mathbf{X}$ and $\mathbf{t}$, respectively at $\mathbf{X} \in S$. Plugging into Equation (134), one gets

$$\left\langle \frac{\partial \hat{c}}{\partial \lambda_i}, \hat{\mathbf{n}} \right\rangle d\hat{s} = \frac{\|\mathbf{X}\|}{Z^3}\sqrt{\frac{\kappa_X \kappa_t}{K}} \left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle ds. \tag{136}$$

The energy in (131) is valid for most functions r_O and r_B used in the literature. However, in this chapter, we chose

$$\begin{aligned} r_O &= -\log(\sigma_O) - \frac{(I(\mathbf{x}) - \mu_O)^2}{\Sigma_O} \quad \text{and} \\ r_B &= -\log(\sigma_B) - \frac{(I(\mathbf{x}) - \mu_B)^2}{\Sigma_B} \end{aligned} \tag{137}$$

as in the region-based active contour technique presented in [93]. Using these functions, the two last terms in Equation (132) collapse for $\mu_{\mathrm{O/B}} = \frac{\int_{R/R^c} I(\hat{\mathbf{x}}) d\Omega}{\int_{R/R^c} d\Omega}$ and $\Sigma_{\mathrm{O/B}} = \frac{\int_{R/R^c} \left(I(\hat{\mathbf{x}}) - \mu_{\mathrm{O/B}}\right)^2 d\Omega}{\int_{R/R^c} d\Omega}$.[3]

Thus, the flow becomes a simple line integral on C

$$\frac{dE}{d\lambda_i} = \int_C \left(r_O\big(I(\pi(\mathbf{X}))\big) - r_B\big(I(\pi(\mathbf{X}))\big) \right) \cdot \frac{\|\mathbf{X}\|}{Z^3} \sqrt{\frac{\kappa_X \kappa_t}{K}} \left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle ds. \tag{138}$$

For $i = 1, 2, 3$ (i.e., λ_i is a translation parameter)

$$\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{R}\mathbf{X_0} + \mathbf{T}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \left\langle \frac{\partial \mathbf{T}}{\partial \lambda_i}, \mathbf{N} \right\rangle = N_i.$$

For $i = 4, 5, 6$ (i.e., λ_i is a rotation parameter), one can similarly show that

$$\left\langle \frac{\partial \mathbf{X}}{\partial \lambda_i}, \mathbf{N} \right\rangle = \langle \mathcal{M}_i, \mathbf{N} \rangle$$

with $\mathcal{M}_i$ the i^{th} column of

$$\mathcal{M} = \mathbf{R} \begin{bmatrix} 0 & Z_0 & -Y_0 \\ -Z_0 & 0 & X_0 \\ Y_0 & -X_0 & 0 \end{bmatrix}$$

.

6.2.4 Implementation remarks

In Equation (138), the computations of the gradients involve the explicit determination of the occluding curve C. From the definition, one can compute

$$C = \{\mathbf{X} \in \mathcal{V}^+ \cap \mathcal{V}^-, \text{ such that } \pi(\mathbf{X}) \in \hat{c}\}$$

[3]For gray-scale images $\mu_{\mathrm{O/B}}$ and $\Sigma_{\mathrm{O/B}}$ are scalars. For color images, $\mu_{\mathrm{O/B}} \in \mathbb{R}^3$ and $\Sigma_{\mathrm{O/B}} \in \mathbb{R}^{3\times 3}$. Texture can also be used, see [91].

where

$$\mathcal{V}^+ = \{\mathbf{X} \in S, \text{ such that } \langle \mathbf{X}, \mathbf{N} \rangle \geq 0\} \ \& \ \mathcal{V}^- = \{\mathbf{X} \in S, \text{ such that } \langle \mathbf{X}, \mathbf{N} \rangle \leq 0\}.$$

N.B.: The set V (respectively, $V^c = S \setminus V$) of points $\mathbf{X} \in S$ that are visible (respectively, not visible) from the camera center is such that $V \subset \mathcal{V}^+$ (respectively, $V^c \supset \mathcal{V}^-$).

The term $\sqrt{\frac{\kappa_X \kappa_t}{K}}$ can be estimated using the (pre-computed) principal curvatures and principal directions for each point $\mathbf{X} \in S$, and the Euler formula (see [33]). To save computational time, however, we used the approximation $\sqrt{\frac{\kappa_X \kappa_t}{K}} \simeq 1$ in our implementation of Equation (138), which still decreased the energy E.

6.3 Experiments

We now report our experimental results obtained for synthetic and real images or sequences. Three different 3D models of complex shapes (see Figure 49) were used to perform segmentation and tracking under diverse scenarios.

6.3.1 Synthetic examples

In the first series of experiments, a sequence of 200 images was constructed by continuously transforming the 3D model of the "CVPR" logo and projecting it into the image plane, using the parameters of a simulated calibrated camera (e.g., focal length f=200). The translation parameters were continuously (but not linearly) varied from 0 to 36 in the X-direction, -100 to 100 in the Y-direction and 475 to 525 in the Z-direction. The rotation axis was also varied as well as the angle. The total angle variation over the sequence exceeded 160°. These ranges of parameters were chosen to ensure a large variation of the aspect of the object.

From the basic sequence obtained, diverse scenarios were tested (See Figure 50). First, the basic sequence was modified by adding Gaussian noise of variance 10% and

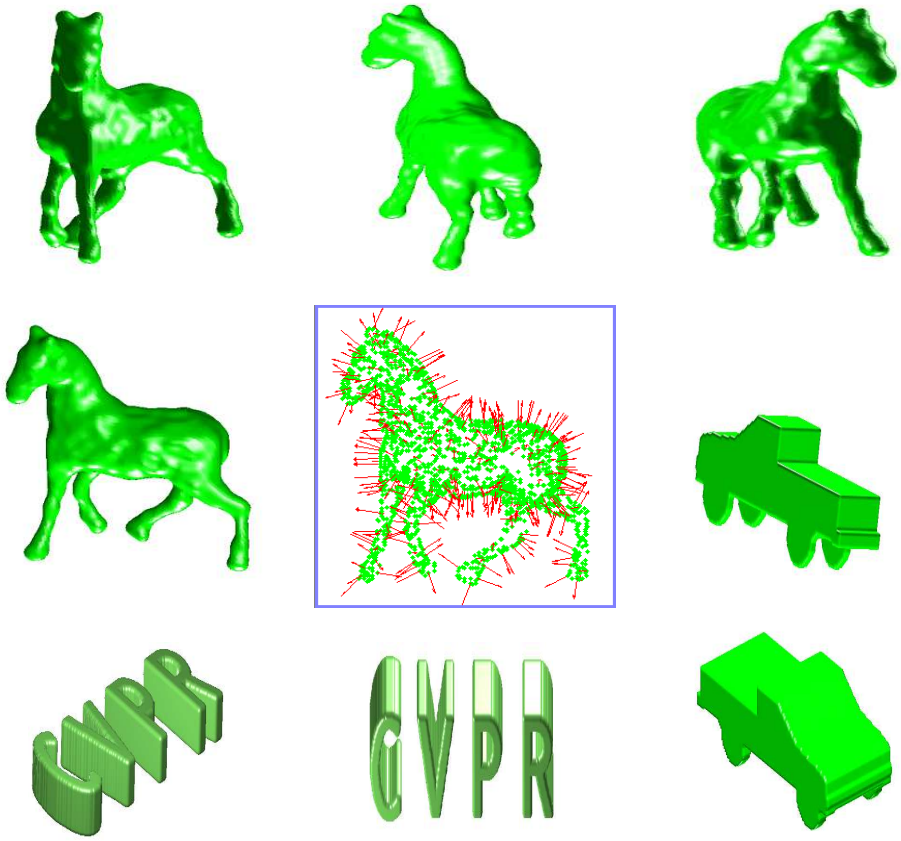


Figure 49: Different views of the 3D models used (rendered surfaces). Framed image: Horse point set with outward normals (sub-sampled).

20%. Second, the basic sequence was modified by adding a rectangular occlusion that could mask more than 1/3 of the logo; Gaussian noise of variance 5% was also added. Third, the basic sequence was modified by adding an occlusion (the word "2008") that masked the object at several places; Gaussian noise of variance 10% was also added. The sequence was tracked using a very simple scheme: For each image, initialization was performed using the pose parameters corresponding to the minimum of the energy obtained for the preceding image and the proposed approach was run until convergence.

Some of the typical visual results obtained are reproduced in Figure 50: From the segmentation point of view, one notes that despite noise and occlusions, accurate segmentations are obtained (in particular missing letters or parts are accurately localized). From the tracking point of view, track was maintained for all scenarios, despite the severe changes in aspect and even topology of the word (e.g., letters initially in contact split). Figures 51 and 52 reproduce the results of the pose estimation procedure. For each image, percent *absolute* errors with respect to the ground-truth were computed for both the translation and rotation as

$$\text{error} = \frac{\|\mathbf{V}_{\text{measured}} - \mathbf{V}_{\text{truth}}\|}{\|\mathbf{V}_{\text{truth}}\|}$$

with $\mathbf{V}$ being a translation or quaternion vector. From the pose estimation point of view, the method appears to perform reasonably well: average error and standard deviation computed over the 200 frames of each sequence rarely exceed 2% and 1%, respectively, for *both* translation and rotation.

In Figure 53, we show segmentation results (and 3D coordinates recovery of the object) for a synthetic color image. Despite the fact that the object is partially occluded, a satisfying segmentation is obtained. In Figure 54, we segment 3 noisy images. These images are entirely composed of Gaussian noise and would pose a real challenge to methods based of local features. The first image is composed of two

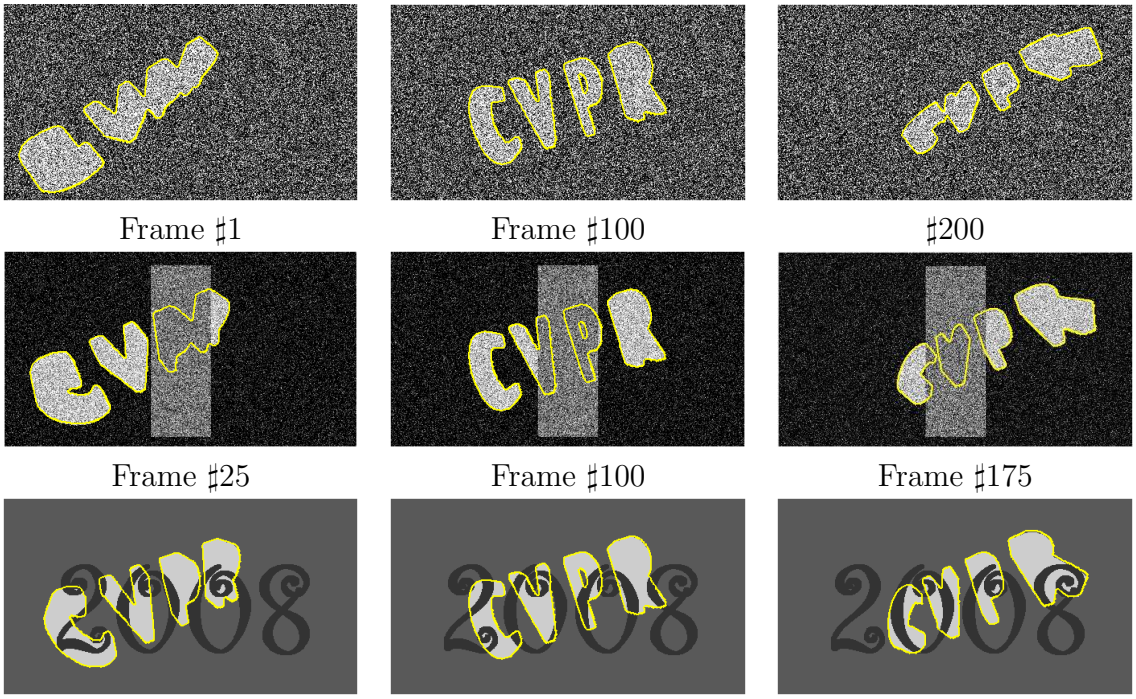


Figure 50: Visual tracking results for the sequences involving the "CVPR" logo (Yellow curves). *First row:* Original sequence with Gaussian noise ($\sigma = 20\%$). *Second row:* Rectangular Occlusion and Gaussian noise ($\sigma = 5\%$). *Third row:* "2008" Occlusion.

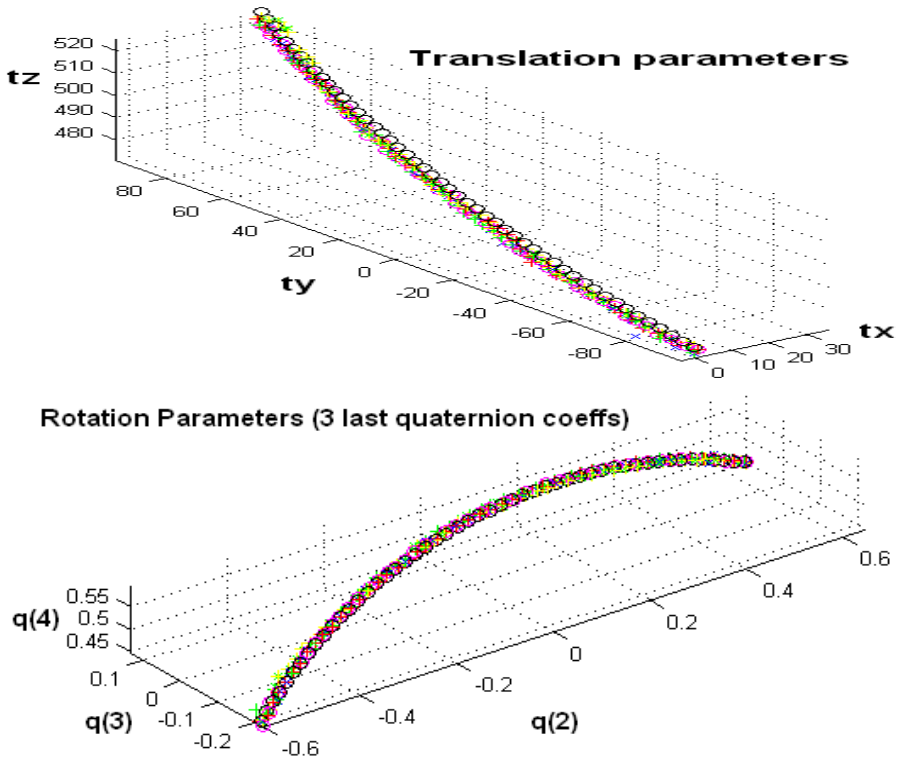


Figure 51: Quantitative tracking results for the "CVPR" sequences. Plot Results for translation and rotation parameters (ground truth: black O).

regions with different intensity means, in grayscale. The second image is composed of two regions with the same mean intensities and different variances ($\mu_O = .5$, $\sigma_O = 10\%$ and $\sigma_B = 20\%$), in grayscale. The third image is composed of two regions with the same mean intensities and different variances, in the color space. Despite initializations that are quite far from the truth, accurate segmentations and pose estimates are obtained.

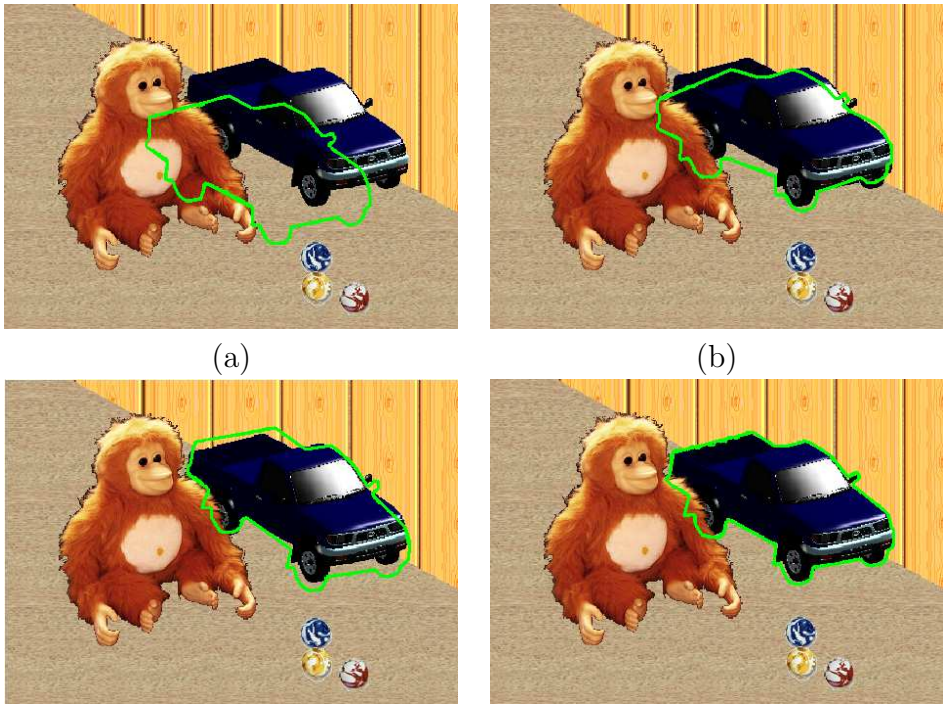
6.3.2 Real scene

Figure 55 shows the tracking results obtained for a real sequence. The sequence is composed of 32 images of a toy horse. The images were taken from discrete positions of a calibrated camera that performed a complete rotation around the object. Tracking this sequence would be challenging for most algorithms available in the literature. For

Scenario ↓	mean error	std. dev. error	max error
occl=N/A $\sigma = 10\%$	Trans:0.7% Rot.:0.7%	Trans:0.07% Rot.:0.44%	Trans:1.0% Rot.:3.6%
occl=N/A $\sigma = 20\%$	Trans:0.8% Rot.:1.4%	Trans:0.13% Rot.:0.69%	Trans:1.4% Rot.:3.2%
occl=N/A $\sigma = 30\%$	Trans:0.9% Rot.:1.2%	Trans:0.21% Rot.:1.19%	Trans:2.0% Rot.:7.3%
occl=rect. $\sigma = 5\%$	Trans:0.7% Rot.:0.6%	Trans:0.12% Rot.:0.33%	Trans:1.1% Rot.:1.5%
occl=2008 $\sigma = 0\%$	Trans:0.6% Rot.:1.2%	Trans:0.20% Rot.:1.02%	Trans:1.6% Rot.:6.6%
occl=2008 $\sigma = 10\%$	Trans:0.6% Rot.:1.4%	Trans:0.21% Rot.:1.08%	Trans:1.06% Rot.:6.1%

Figure 52: Quantitative tracking results for the "CVPR" sequences. Table recapitulating %-**absolute** error statistics over the 200 images composing each sequence.

instance, 3D pose estimation techniques based on local features such as points or edges (e.g., [97, 75]) could be thrown off by the textured/noisy background (false features). In addition, the shape of the object is very complex and impossible to describe in terms of geometric primitives such as lines, ellipses, etc. ([32, 75]) or even algebraically ([104, 36]). Furthermore, the aspect of the object changes drastically from one frame to the next. This is due notably to the numerous self-occlusions that occur throughout the sequence (e.g., the elongated limbs of the animal often get occluded by its body). Finally, the camera "jumps" between successive images creating large changes in the pose of the object that need to be recovered (e.g., changes in the angular position of the camera can exceed 15° between frames). Despite these difficulties, very satisfying tracking performances were observed using our approach and the tracking scheme alluded to above. This highlights the robustness of the technique to initialization since the large cameras jumps can be accommodated. We note that region-based active contour techniques such as [12], would also lead to satisfying segmentations on this particular sequence. However, these approaches would not also determine the pose of the object, which is valuable information for tracking applications.



(a) (b)

(c) (d)

Figure 53: Applicability of the method on color images (note also the partial occlusion). (a): Initialization; (b) and (c): Intermediate steps of the evolution; (d): Final result.

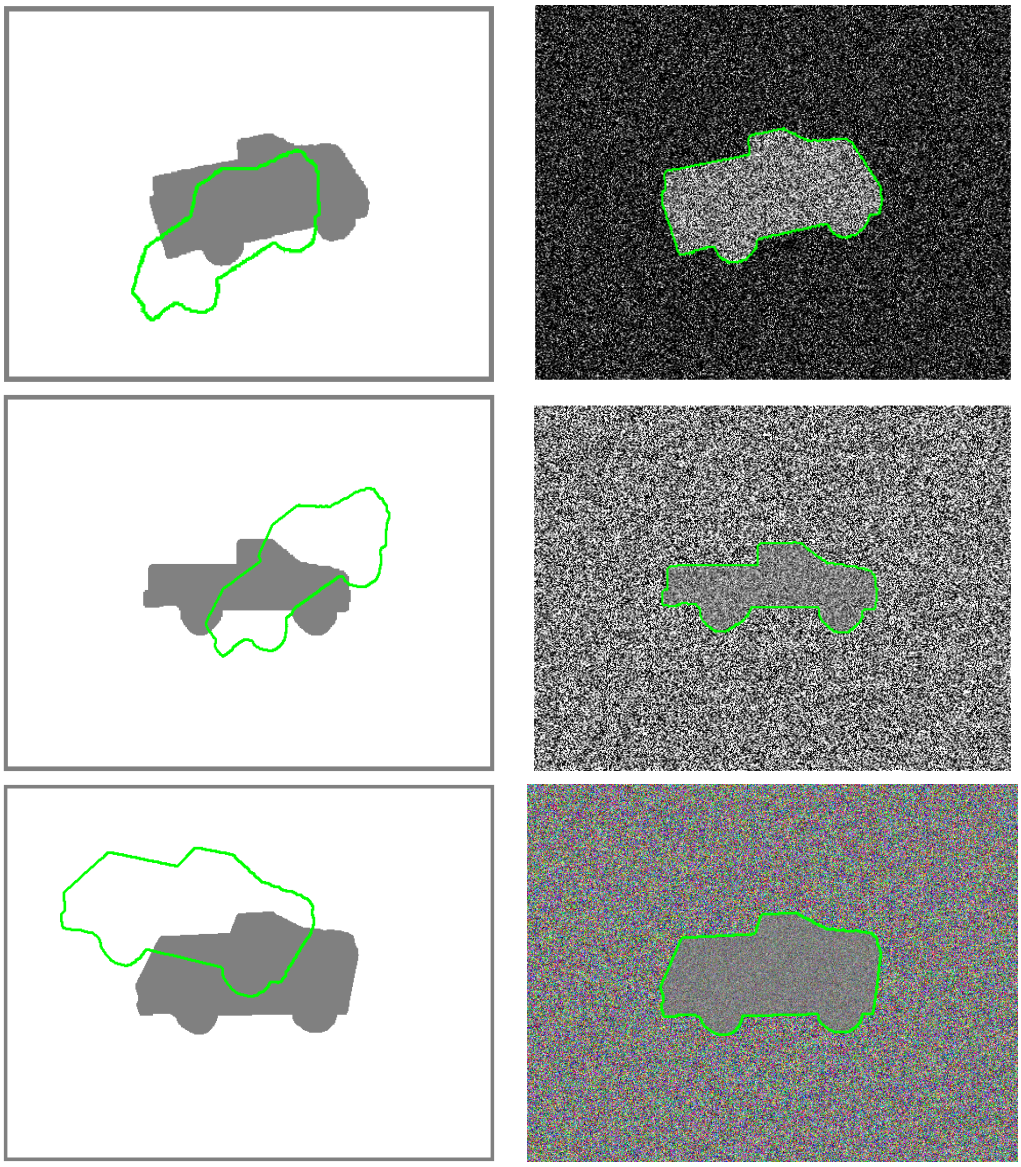


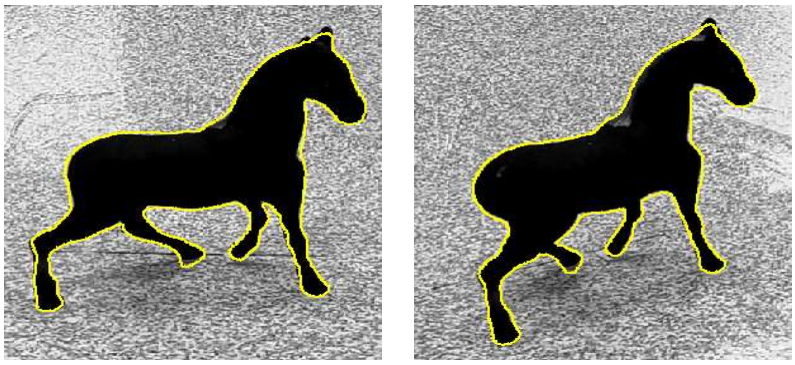
Figure 54: Applicability of the method on noisy images (Gaussian noise). *Left*: Original regions and initializations. *Right*: Results. *First row:* Grayscale Image composed of two regions with different intensity averages; *Second row:* Grayscale Image composed of two regions with same intensity averages and different variances; *Third row:* Color Image composed of two regions with same intensity averages and different variances.

To further test the robustness of the method to missing information, we used images extracted from the horse sequence above and occluded different parts of the horse body (notably the legs that have valuable information about the angular position of the horse). Diverse pose parameters quite far from the truth were used as initializations (e.g., angular position could be off by 20°), and our algorithm was run until convergence. Despite the occlusions (with various pixel intensities or texture) and poor initializations, very convincing segmentations were obtained; see Figure 56. Also, the positions of the object in the camera referential were accurately recovered. As can be noticed by comparing with Figure 55, the results in the presence of occlusion are very comparable to the ones without occlusion. These experiments would pose a challenge to most region-based active contour techniques, even using shape priors [26, 70], because of the large catalogue of 2D shapes that would need to be learnt to achieve similar performances.

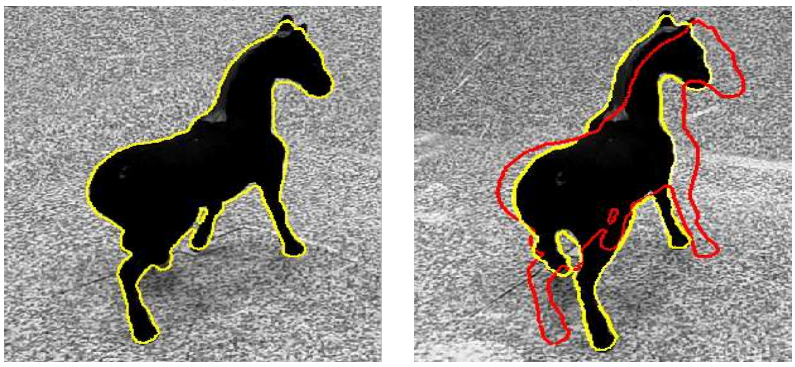
6.4 Chapter Conclusion

In this work, we presented a region-based approach to the 3D pose estimation problem. This approach differs from other 3D pose estimation algorithms since it does not rely on local image features. Our method extends the framework of [134], and allows one to employ global image statistics to drive the pose estimation process. This confers a satisfying level of robustness to noise and initialization to our model as well as bypasses the need to establish correspondences between image and object features.

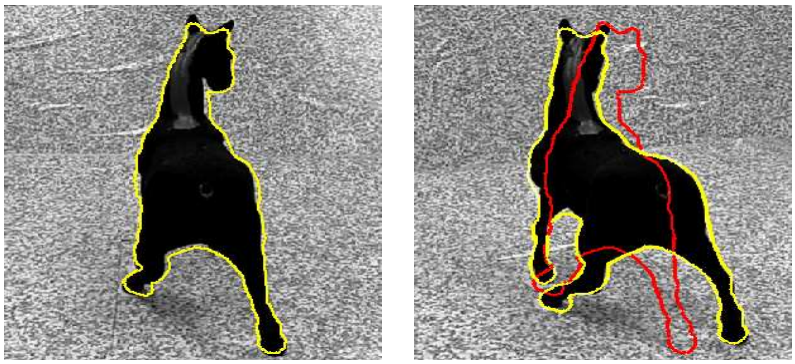
Furthermore, the approach possesses the typical qualities of a region-based active contour technique with shape prior, such as robustness to occlusion or missing information, without the need to evolve an infinite dimensional contour. Also, the prior knowledge of the shape of the object is compactly represented by a unique 3D model. The main advantage of applying the proposed technique is that it results not only in locating the object in the 2D image (a typical task handled by GAC approaches) but



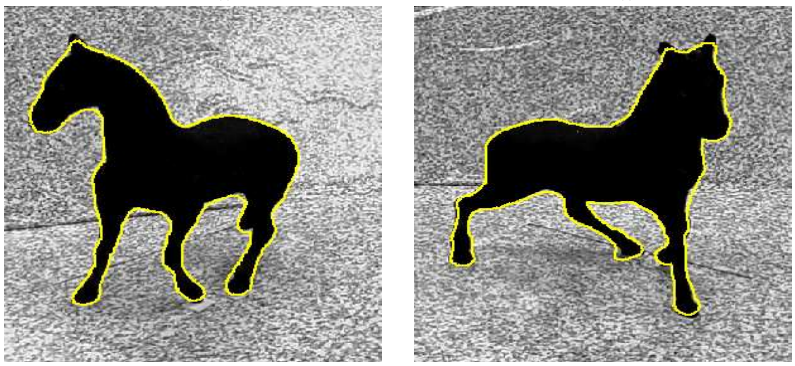
Frame #1 Frame #4



Frame #5 Frame #6

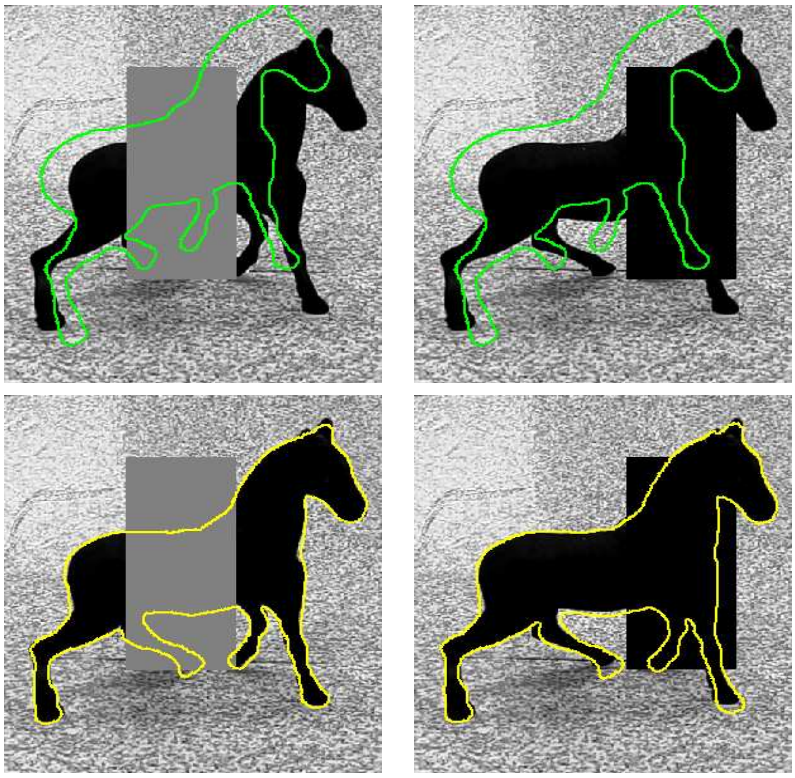


Frame #9 Frame #10

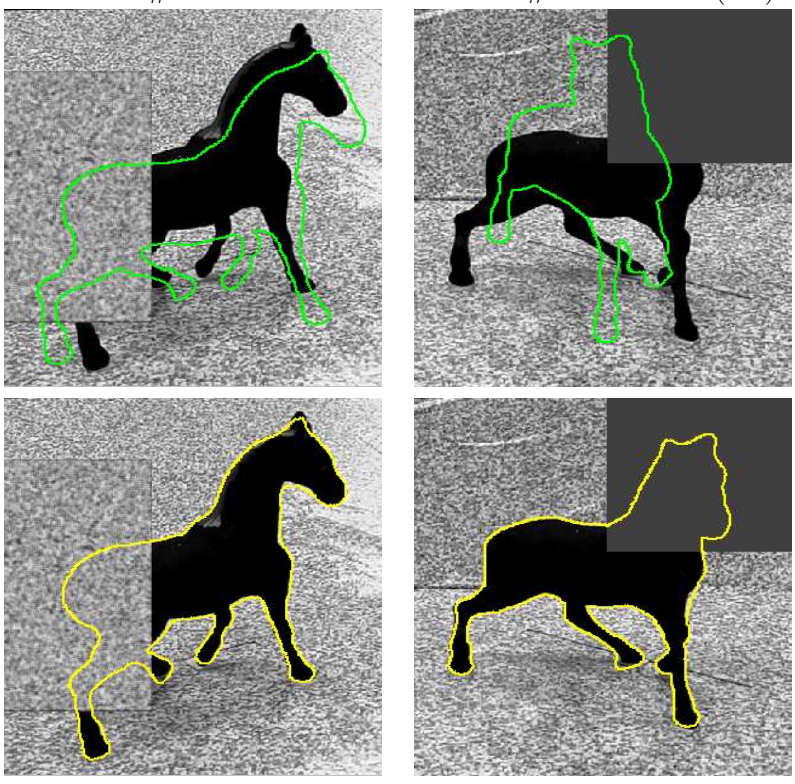


Frame #19 Frame #29

Figure 55: Tracking results for the horse sequence *Yellow contours*: final results after convergence; *Red contours*: Initializations from the result of the preceding image (see text for our tracking scheme). The aspect of the object changes drastically throughout the sequence. The position of the object is "very discontinuous" between successive images.



Frame #1 Occluded | Frame #1 Occluded (bis)



Frame #4 Occluded | Frame #29 Occluded

Figure 56: Segmentation results with occlusions. *Green contours*: Some of the initializations tested; *Yellow contours*: Final results after convergence (these results are almost identical to the ones obtained in Figure 55 with no occlusion).

also in the world (a typical task handled by 2D-3D pose estimation algorithms). This makes the method particularly suitable for tracking applications involving a unique calibrated camera.

A possible direction for future research is to extend the proposed approach to include the knowledge of multiple 3D shapes. In particular, the method in [124] (where evolution of parameters in the shape space is performed in addition to pose parameters) could be adapted to the problem at hand. It is expected that the resulting framework will allow to learn the possible deformations of the object and lead to robust performances for non-rigid registration and tracking tasks.

CHAPTER VII

CURVE SHORTENING ON A SURFACE WITH APPLICATIONS TO STOCHASTIC SNAKES

Curvature driven flows have been extensively considered from a deterministic point of view. In this chapter, we describe a stochastic approximation of curve shortening flows valid for arbitrary embedded curves on a Riemannian surface. A stochastic interpretation of geodesic active contours of interest in computer vision follows from this framework. This chapter is organized as follows: In Section 7.2, we write down the basic evolution equations for curve shortening on a surface in a form which will fit our stochastic approximation techniques. In Section 7.3, we specialize our set-up to the conformal Euclidean case (for the geodesic snakes). In Section 7.4, we describe the stochastic model. In Section 7.5, we present a few implementation considerations. Finally, in Section 7.6, we present experiments highlighting some aspects of our stochastic curve evolution method.

7.1 Motivation and Related Work

In this chapter, we describe a model of stochastic snakes based on the theory of interacting particle systems. In some previous work [5], a stochastic interpretation of curve shortening flows was described. This brought together the theories of curve evolution and hydrodynamical limits, and as such impacted on the growing use of joint methods from probability and partial differential equations (PDE's) in the image processing and computer vision. In this work, we will indicate how this theory

may be implemented to forge a novel stochastic curve shortening algorithm on a Riemannian surface, for non-convex curves. As a corollary, we will formulate a stochastic approximation of geodesic active contours, which are used for problems in computer vision [10, 63].

The basic idea is to consider curve shortening according to geodesic curvature on a Riemannian surface as in [54, 55], and apply the techniques of [3] to the case at hand. Indeed, previous work [3] described a stochastic approximation for curve shortening for planar curves. For a Riemannian surface, the Gaussian curvature plays an important role, and so in the end our model involves a zero range birth/death process with drift. Finally, taking a conformally Euclidean metric in the plane, we derive a stochastic model of the conformal (geodesic) active contours from [10, 63].

7.2 Evolution equations

Unless otherwise stated in this section, $\mathcal{M}$ denotes a smooth Riemannian surface equipped with Riemannian metric g which is convex at infinity. All lengths will be computed with respect to g. We consider a family of parameterized curves $C : \mathcal{M} \times [0, T) \to \mathbb{R}^2$ which evolves "normally,"

$$\frac{\partial C}{\partial t} = v\mathbf{N}, \tag{139}$$

where v is the normal velocity of the curve and $\mathbf{N}$ is the unit normal. The curve is supposed to be parameterized by $p \in S^1$ (S^1 is the unit circle). We write κ for the geodesic curvature of C, and

$$\mathbf{T} = \frac{C_p}{\|C_p\|} \tag{140}$$

for its unit tangent. Note that the geodesic curvature κ is defined by

$$\nabla_T \mathbf{T} = \kappa \mathbf{N}. \tag{141}$$

Here,we derive equations for the evolution of the geodesic curvature κ, as a function of the re-normalized arclength. First, the arclength function is defined as

$$s(p,t) = \int_0^p \|C_p(p',t)\| \mathrm{d}p', \tag{142}$$

From [54, 55], we have that

$$\frac{\partial \kappa}{\partial t} = v_{ss} + \big(\kappa^2 + K(C(s,t)\big)v. \tag{143}$$

For Curve Shortening, where $v = \kappa$, this becomes

$$\frac{\partial \kappa}{\partial t} = \kappa_{ss} + \big(\kappa^2 + K(C(s,t)\big)\kappa. \tag{144}$$

Here, time and space derivatives do not commute. Instead, one has

$$[\partial_t, \partial_s] = \kappa v \partial_s = \kappa^2 \partial_s. \tag{145}$$

We introduce the total length

$$L(t) = s(1,t) = \int_0^1 \|C_p(p',t)\| \mathrm{d}p'. \tag{146}$$

Then

$$\frac{\partial s}{\partial t} = -\int_0^s \kappa^2 \mathrm{d}s \text{ and } L'(t) = -\int_0^{L(t)} \kappa^2 \mathrm{d}s. \tag{147}$$

We can now regard C and κ as functions of s instead of p, which gives us

$$\begin{aligned} \left.\frac{\partial \kappa}{\partial t}\right|_s &= \left.\frac{\partial \kappa}{\partial t}\right|_p - \left.\frac{\partial s}{\partial t}\right|_p \frac{\partial \kappa}{\partial s} \\ &= \kappa_{ss} + \big(\kappa^2 + K(C)\big)\kappa + \kappa_s \int_0^s \kappa^2 \mathrm{d}s \\ &= \frac{\partial}{\partial s}\left\{\kappa_s + \kappa \int_0^s \kappa^2 \mathrm{d}s\right\} + K(C(s,t))\kappa. \end{aligned} \tag{148}$$

The arclength does not have a fixed domain. To fix this domain, one considers the normalized arclength

$$z := \frac{s}{L(t)}, \tag{149}$$

and one introduces

$$\hat{\kappa}(z,t) = L(t)\kappa(s,t). \tag{150}$$

Then, one has

$$\hat{\kappa}\mathrm{d}z = \kappa \mathrm{d}s, \quad \kappa_s = L(t)^{-2}\hat{\kappa}_z, \quad \kappa_{ss} = L(t)^{-3}\hat{\kappa}_{zz}, \tag{151}$$

and the rescaled curvature $\hat{\kappa}$ satisfies

$$\begin{aligned}\frac{\partial \hat{\kappa}}{\partial t} &= L'(t)\kappa(s,t) + L(t)\frac{\partial \kappa}{\partial t} \\ &= -\kappa(s,t)\int_0^{L(t)} \kappa^2 \mathrm{d}s + L\frac{\partial \kappa}{\partial t} \\ &= -L(t)^{-2}\int_0^1 \hat{\kappa}^2 \mathrm{d}z' + \frac{\partial \kappa}{\partial t}.\end{aligned}$$

Using (148), one finds

$$L(t)^2\frac{\partial \hat{\kappa}}{\partial t} = \frac{\partial}{\partial z}\left\{\hat{\kappa}_z + \hat{\kappa}\big(\int_0^z \hat{\kappa}^2 \mathrm{d}z' - z\int_0^1 \hat{\kappa}^2 \mathrm{d}z'\big)\right\} + L(t)^2 K(C(s,t))\hat{\kappa} \tag{152}$$

in which one can set

$$b_{\hat{\kappa}}(z,t) := \big(\int_0^z \hat{\kappa}^2 \mathrm{d}z' - z\int_0^1 \hat{\kappa}^2 \mathrm{d}z'\big).$$

We can now set up a closed system of evolution equations after introducing a new time variable τ, which satisfies $\mathrm{d}\tau = L^{-2}\mathrm{d}t$,

$$\frac{\partial \hat{\kappa}}{\partial \tau} = \hat{\kappa}_{zz} + (b_{\hat{\kappa}}\hat{\kappa})_z + L^2 K(C)\hat{\kappa} \tag{153}$$

$$\frac{\mathrm{d}L}{\mathrm{d}\tau} = -L\int_0^1 \hat{\kappa}^2 \mathrm{d}z \tag{154}$$

$$\frac{\mathrm{d}t}{\mathrm{d}\tau} = L^2 \tag{155}$$

$$\frac{\partial C}{\partial \tau} = L^2\kappa\mathbf{N} = L\hat{\kappa}\mathbf{N} \tag{156}$$

Here, one could let $\hat{\kappa}$ evolve stochastically (by a zero range birth-death process with drift), and after each step update L and C. In the conformally Euclidean case (see below) one could use the given curve parametrization to compute the Gauss curvature K from (172) as well as the unit normal N from (168).

To obtain a system which only contains positive or nonnegative densities one sets $\hat{\kappa} = \lambda - \mu$, and lets λ and μ both evolve by the following system

$$\frac{\partial \lambda}{\partial \tau} = \lambda_{zz} + (b_{\lambda-\mu}\lambda)_z + L^2 K(C)\lambda \tag{157}$$

$$\frac{\partial \mu}{\partial \tau} = \mu_{zz} + (b_{\lambda-\mu}\mu)_z + L^2 K(C)\mu \tag{158}$$

$$\frac{\mathrm{d}L}{\mathrm{d}\tau} = -L\int_0^1 (\lambda-\mu)^2 \mathrm{d}z' \tag{159}$$

$$\frac{\mathrm{d}t}{\mathrm{d}\tau} = L^2 \tag{160}$$

$$\frac{\partial C}{\partial \tau} = L(\lambda-\mu)\mathbf{N} \tag{161}$$

in which

$$b_{\lambda-\mu}(z,\tau) = \int_0^z \{\lambda(z',\tau) - \mu(z',\tau)\}^2 \,\mathrm{d}z' - z\int_0^1 \{\lambda(z',\tau) - \mu(z',\tau)\}^2 \,\mathrm{d}z' \tag{162}$$

and

$$(b_{\lambda-\mu})_z = (\lambda-\mu)^2 - \int_0^1 (\lambda-\mu)^2 \mathrm{d}z' \tag{163}$$

is simply the average free part of $(\lambda-\mu)^2$.

The maximum principle will keep both λ and μ positive, as long as $\lambda(\cdot,t_0), \mu(\cdot,t_0) \geq 0$. For any given initial $\hat{\kappa}(\cdot,t_0)$ we can take

$$\lambda(z,t_0) = \max(\hat{\kappa}(z,t_0),0), \quad \mu(z,t_0) = \max(-\hat{\kappa}(z,t_0),0)$$

(which have disjoint support), or we can take

$$\mu(z,t_0) = \mu = -\min_{0\leq z\leq 1}(\hat{\kappa}(z,t_0),0), \quad \lambda(z,t_0) = \mu + \hat{\kappa}(z,t_0).$$

7.3 Formulas in Conformal Euclidean Case

In this section, we set $M = \mathbb{R}^2$ with Riemannian metric

$$g = e^{2u(x,y)}\big\{(\mathrm{d}x)^2 + (\mathrm{d}y)^2\big\}. \tag{164}$$

As above, all lengths are measured using the metric g, so, if $C(p,t) = \begin{pmatrix} x(p,t) \\ y(p,t) \end{pmatrix}$ then

$$C_p = \begin{pmatrix} x_p(p,t) \\ y_p(p,t) \end{pmatrix} \tag{165}$$

$$\|C_p\| = e^{u(x(p,t),y(p,t))}\sqrt{x_p(p,t)^2 + y_p(p,t)^2} \tag{166}$$

$$\mathbf{T} = \frac{e^{-u(x(p,t),y(p,t))}}{\sqrt{x_p(p,t)^2 + y_p(p,t)^2}} \begin{pmatrix} x_p(p,t) \\ y_p(p,t) \end{pmatrix} \tag{167}$$

$$\mathbf{N} = \frac{e^{-u(x(p,t),y(p,t))}}{\sqrt{x_p(p,t)^2 + y_p(p,t)^2}} \begin{pmatrix} -y_p(p,t) \\ x_p(p,t) \end{pmatrix} \tag{168}$$

The velocity vector is

$$\frac{\partial C}{\partial t} = v\mathbf{N} = \begin{pmatrix} x_t(p,t) \\ y_t(p,t) \end{pmatrix},$$

so that the normal velocity is

$$v = g\big(\frac{\partial C}{\partial t}, \mathbf{N}\big) = e^{u(x(p,t),y(p,t))}\frac{y_t x_p - x_t y_p}{\sqrt{x_p(p,t)^2 + y_p(p,t)^2}}.$$

As above, the geodesic curvature κ is defined by

$$\nabla_{\mathbf{T}}\mathbf{T} = \kappa\mathbf{N}. \tag{169}$$

To compute it, we introduce the orthonormal frame

$$X = e^{-u}\frac{\partial}{\partial x}, \qquad Y = e^{-u}\frac{\partial}{\partial y}.$$

For this frame, one has

$$\nabla X = \alpha \otimes Y, \qquad \nabla Y = -\alpha \otimes X$$

for some one-form α. This form can be computed from the commutator

$$[X,Y] = e^{-u}\big\{u_y X - u_x Y\big\}$$

and the fact that the connection ∇ is torsion free, i.e.

$$[X, Y] = \nabla_X(Y) - \nabla_Y(X) = -\alpha(X)X - \alpha(Y)Y.$$

We see that

$$\alpha(X) = -e^{-u}u_y, \qquad \alpha(Y) = e^{-u}u_x,$$

whence

$$\alpha = -u_y \mathrm{d}x + u_x \mathrm{d}y.$$

In terms of the frame $\{X, Y\}$, one has

$$T = \frac{x_p X + y_p Y}{\sqrt{x_p^2 + y_p^2}}, \qquad N = \frac{-y_p X + x_p Y}{\sqrt{x_p^2 + y_p^2}}.$$

A computation then leads to the following formula for the geodesic curvature

$$\kappa = \frac{e^{-u}}{\sqrt{x_p^2 + y_p^2}} \left\{ \frac{x_p y_{pp} - y_p x_{pp}}{x_p^2 + y_p^2} + u_x y_p - u_y x_p \right\} \tag{170}$$

Compared with the EUCLIDEAN curvature

$$\kappa_E = \frac{x_p y_{pp} - y_p x_{pp}}{\left(x_p^2 + y_p^2\right)^{3/2}}$$

of the curve one has

$$\begin{aligned} \kappa &= e^{-u} \left\{ \kappa_E + \frac{u_x y_{pp} - u_y x_{pp}}{\sqrt{x_p^2 + y_p^2}} \right\} \\ &= e^{-u}(\kappa_E + \nabla u \cdot \mathbf{N}) \end{aligned}$$

(where N denotes the ordinary Euclidean unit normal).

For a metric of the form

$$(\mathrm{d}s)^2 = e^{2u(x,y)}\big((\mathrm{d}x)^2 + (\mathrm{d}y)^2\big) \tag{171}$$

the Gauss curvature K is given by

$$K = -e^{-2u}\Delta u = -e^{-2u}\big(u_{xx} + u_{yy}\big). \tag{172}$$

7.4 Stochastic Interpretation

Since we are interested in a stochastic interpretation, we consider the evolution of a "density" corresponding to Equations (157, 158). Accordingly, using parametrization σ, we interpret the quantities λ and μ as densities. The techniques here follow our previous work in [3] except now we also have a birth/death term.

The approximations we use are based on so-called *interacting particle systems* as described in [66]. Notice that because of our special parametrization the diffusion terms of equations (157) and (158) are linear. In our case, there will be two types of particles, one simulating the λ, one the μ, with the interaction being through the drift and birth/death rates.

Let $T_N = \mathbb{Z}/N\mathbb{Z}$ denote the discrete torus. The configuration of particles at time τ is given by the pair of functions $(\eta_\tau^\lambda(\cdot), \eta_\tau^\mu(\cdot)) : T_N \to \mathbb{N}^2$, and the construction is such that $(\eta_\tau^\lambda([\sigma N]), \eta_\tau^\mu([\sigma N]))$ converges to $(\lambda(\sigma,\tau), \mu(\sigma,\tau))$. Let $g : \mathbb{N} \to \mathbb{R}_+$ (the *diffusion rates*, with $g(0) = 0$), $b^\mu, b^\lambda : T_N \times E_N \to \mathbb{R}_+$ (the *birth rates*) $d^\mu, d^\lambda, : T_N \times E_N \to \mathbb{R}_+$ (the death rates), $h^\lambda, h^\mu : T_N \times \mathbb{R}_+^{\mathbb{N}^2} \to \mathbb{R}$ (the *drift rate*) be given, and define the Markov generator on the particle configuration $E_N = \mathbb{N}^{T_N} \times \mathbb{N}^{T_N}$ by

$$(\mathcal{L}^N f)(\eta^\lambda, \eta^\mu) = N^2(\mathcal{L}_0 f)(\eta^\lambda, \eta^\mu) + N(\mathcal{L}_1 f)(\eta^\lambda, \eta^\mu) + (\mathcal{L}_2 f)(\eta^\lambda, \eta^\mu)\,, \quad f \in C_b(E_N)\,,$$

where

$$\begin{aligned}(\mathcal{L}_0 f)(\eta^\lambda, \eta^\mu) &= \frac{1}{2}\sum_{i\in T_N} g(\eta^\lambda(i))\left[f(\eta^{i,i+1,\lambda}, \eta^\mu) + f(\eta^{i,i-1,\lambda}, \eta^\mu) - 2f(\eta^\lambda, \eta^\mu)\right] \\ &\quad + \frac{1}{2}\sum_{i\in T_N} g(\eta^\mu(i))\left[f(\eta^\lambda, \eta^{i,i+1,\mu}) + f(\eta^\lambda, \eta^{i,i-1,\mu}) - 2f(\eta^\lambda, \eta^\mu)\right]\end{aligned}$$

and

$$\begin{aligned}(\mathcal{L}_1 f)(\eta^\lambda, \eta^\mu) &= \sum_{i\in T_N} h^\lambda(i, \eta^\lambda, \eta^\mu)\left[f(\eta^{i,i+\mathrm{sign}(h^\lambda(i,\eta^\lambda,\eta^\mu)),\lambda}, \eta^\mu) - f(\eta^\lambda, \eta^\mu)\right], \\ &\quad + \sum_{i\in T_N} h^\mu(i, \eta^\lambda, \eta^\mu)\left[f(\eta^\lambda, \eta^{i,i+\mathrm{sign}(h^\mu(i,\eta^\lambda,\eta^\mu)),\mu}) - f(\eta^\lambda, \eta^\mu)\right],\end{aligned}$$

$$
\begin{aligned}
(\mathcal{L}_2 f)(\eta^\lambda, \eta^\mu) &= \sum_{i \in T_N} b^\lambda(i, \eta^\lambda, \eta^\mu) \left[f(\eta^{i,+,\lambda}, \eta^\mu) - f(\eta^\lambda, \eta^\mu) \right] \\
&\quad + \sum_{i \in T_N} d^\lambda(i, \eta^\lambda, \eta^\mu) \left[f(\eta^{i,-,\lambda}, \eta^\mu) - f(\eta^\lambda, \eta^\mu) \right] \\
&\quad + \sum_{i \in T_N} b^\mu(i, \eta^\lambda, \eta^\mu) \left[f(\eta^\lambda, \eta^{i,+,\mu}) - f(\eta^\lambda, \eta^\mu) \right] \\
&\quad + \sum_{i \in T_N} d^\mu(i, \eta^\lambda, \eta^\mu) \left[f(\eta^\lambda, \eta^{i,-,\mu}) - f(\eta^\lambda, \eta^\mu) \right] ,
\end{aligned}
$$

where, with $\star \in \{\lambda, \mu\}$,

$$
\eta^{i,i\pm1,\star}(j) = \begin{cases} \eta^\star(j) + 1, & j = i \pm 1, \eta^\star(i) \neq 0, \\ \eta^\star(j) - 1, & j = i, \eta^\star(i) \neq 0, \\ \eta^\star(j), & \text{else} \end{cases}
$$

$$
\eta^{i,+,\star}(j) = \begin{cases} \eta^\star(j) + 1, j = i, \\ \eta^\star(j), \text{else} \end{cases}
$$

$$
\eta^{i,-,\star}(j) = \begin{cases} \eta^\star(j) - 1, j = i, \eta^\star(i) > 0, \\ \eta^\star(j), \text{else} \end{cases} .
$$

Note that $\mathcal{L}_0$ will be used to approximate the diffusion term of equations (157, 158), $\mathcal{L}_1$ will be used to approximate the drift term, while the birth/death part $\mathcal{L}_2$ will approximate the reaction term of (157,158).

A slightly non-standard feature of the system (157, 158) is that the drift and reaction terms are *non-local*. We proceed however with the standard ansatz, namely that because the diffusion term is constant, the local equilibrium measure for the particle system is a product of Poisson measures. A full justification of this ansatz, following closely the derivation in [82], does require some work and will be detailed elsewhere.

Let X be Poisson random variable X of parameter α, so that $EX = \alpha$. Thus, referring to Equation (163), to define the rates, set the function $B : E_N \times T_N \to \mathbb{R}_+$

as

$$B(\eta^\lambda,\eta^\mu)(j) = \frac{1}{N}\sum_{i=1}^{j}\left([\eta^\lambda(i)-\eta^\mu(i)]^2-[\eta^\lambda(i)+\eta^\mu(i)]-c\right),$$
$$c = \frac{1}{N}\sum_{i=0}^{N-1}\left([\eta^\lambda(i)-\eta^\mu(i)]^2-[\eta^\lambda(i)+\eta^\mu(i)]\right). \quad (173)$$

Note that if $\eta^\lambda(j),\eta^\mu(j)$ are taken as independent Poisson variables of rates $\lambda(j/N,\tau),\mu(j/N,\tau)$, respectively, then

$$E(B(\eta^\lambda,\eta^\mu)(j)) = b_{\lambda(j/N,\tau)-\mu(j/N,\tau)}\,.$$

Following [66, 5], for (157), the rates can be taken as follows:

$$g(k)=k,$$

$$h^\lambda(i,\eta^\lambda,\eta^\mu) = -B(\eta^\lambda,\eta^\mu)(i)\eta^\lambda(i), \quad h^\mu(i,\eta^\lambda,\eta^\mu) = -B(\eta^\lambda,\eta^\mu)(i)\eta^\mu(i),$$
$$b^\lambda(i,\eta^\lambda,\eta^\mu) = \begin{cases} L^2K(C)\eta^\lambda(i) & \text{for } K(C)\geq 0,\\ 0 \quad \text{otherwise,}\end{cases}$$
$$d^\lambda(i,\eta^\lambda,\eta^\mu) = \begin{cases} -L^2K(C)\eta^\mu(i) & \text{for } K(C)\leq 0,\\ 0 \quad \text{otherwise,}\end{cases},$$

and similarly for $b^\mu(i,\eta^\lambda,\eta^\mu), d^\mu(i,\eta^\lambda,\eta^\mu)$.

Note that we take the minus sign in the drift since we have a forward diffusion. The rate for g is of course classical, and the drift rate h is similar to [66], when one takes into account the extra averaging due to the non-local nature of the function $b_{\lambda-\mu}$.

7.5 Implementation

We now present implementation details, allowing one to perform the stochastic curvature evolution of a general (non-convex) contour, in the particular case where the Riemannian surface is flat, i.e. $K=0$ (Note that the case dealing only with convex curves was presented in [126], adopting a quite different approach).

In this particular case, we have 4 possible events that can occur at site i, for a system in state (η^λ, η^μ) in $\mathbb{N}^{T_N} \times \mathbb{N}^{T_N}$ at time t:

- $g^\lambda(i, \eta^\lambda, \eta^\mu)$ = Diffusion of particles λ, which corresponds to a *probabilistic* jump to right or left (each occurring with probability $\frac{1}{2}$),
- $h^\lambda(i, \eta^\lambda, \eta^\mu)$ = Drift of particles λ, which corresponds to a *deterministic* jump to right or left (depending on $\text{sign}(h^\lambda(i, \eta^\lambda, \eta^\mu))$),
- $g^\mu(i, \eta^\lambda, \eta^\mu)$ = Diffusion of particles μ, which corresponds to a *probabilistic* jump to right or left (each occurring with probability $\frac{1}{2}$),
- $h^\mu(i, \eta^\lambda, \eta^\mu)$ = Drift of particles μ, which corresponds to a *deterministic* jump to right or left (depending on $\text{sign}(h^\mu(i, \eta^\lambda, \eta^\mu))$).

7.5.1 Simulation of random events

Let

$$\begin{aligned} E_i &\doteq \left\{N^2 g^\lambda(i, \eta^\lambda, \eta^\mu), N^2 g^\mu(i, \eta^\lambda, \eta^\mu), N h^\lambda(i, \eta^\lambda, \eta^\mu), N h^\mu(i, \eta^\lambda, \eta^\mu)\right\} \\ &= \left\{e_1^i(\eta), e_2^i(\eta), e_3^i(\eta), e_4^i(\eta)\right\} \end{aligned} \tag{174}$$

denote the set of possible transition rates for the system in state $\eta = (\eta^\lambda, \eta^\mu)$ at site i. One can choose a possible event by its rate $e_j^i \in E_i$. The total rate at a given site i can be computed, summing the rates of all possible events, as

$$U(\eta, i) = U\left(\eta^\lambda, \eta^\mu, i\right) \doteq \sum_{j=1}^{4} e_j^i \tag{175}$$

The Poisson system that simulates the Markov process described above can be implemented as

1. Get values for

$$T_i \sim e^{(U(\eta, i))}.$$

which means that T_i's are exponential random variables with parameter $U(\eta, i)$.

2. Set

$$T \doteq \min_{i \in T_N} \{T_i\} = T_{i^*}$$

where i^* is the site where the transition occurs at time $t + T$

3. Take $e_j^i \in E_i$ with the (conditional) transition probability

$$p_{i,j}(\eta) = \frac{e_j^i(\eta)}{U(\eta, i)}$$

to determine the event in E_i that actually occur at time $t + T$.

7.5.2 Curve Reconstruction

This section deals with the reconstruction of a non-convex curve, given its scaled curvature. This reconstruction is possible, since a curve parameterized by its arc-length s is fully determined by its curvature field $\kappa(s)$, up to a rigid transformation (rotation and translation).

Let $C : [0, L] \mapsto \mathbf{R}^2$ be a curve parameterized by its arc-length s (L denotes the total length of the curve). Let $\theta(s)$ be the angle between the unit tangent $\mathbf{T}(s)$ at a point $(x(s); y(s))$ of C and the positive direction of the x-axis. Let $\kappa(s)$ denote the curvature of C at point $(x(s); y(s))$. By definition of the curvature, one has

$$\theta(s) = \int_0^s \kappa(\tilde{s}).d\tilde{s} + \varphi_0, \tag{176}$$

where φ_0 is a constant determining the general orientation of the curve. The unit tangent to the curve can of course be expressed as $\mathbf{T}(s) = (\cos\theta(s); \sin\theta(s))$.

Re-parameterizing the curve using the normalized arc-length $z = \frac{s}{L}$, the curve C becomes $C : S^1 \mapsto \mathbf{R}^2$. One can then introduce the scaled curvature

$$\hat{\kappa}(z) = L\kappa(s). \tag{177}$$

From (176) and remarking that $\hat{\kappa}dz = \kappa ds$, one then has

$$\theta(Lz) = \int_0^z \hat{\kappa}(\tilde{z}).d\tilde{z} + \varphi_0, \tag{178}$$

and the unit tangent to the curve can be reexpressed as $\mathbf{T}(Lz) = (\cos\theta(Lz); \sin\theta(Lz))$ or, more compactly, as

$$\mathbf{T}(Lz) = e^{i\theta(Lz)} = e^{i\{\int_0^z \hat{\kappa}(\tilde{z}).d\tilde{z}+\varphi_0\}}. \tag{179}$$

Noticing that $\frac{dC}{ds} = \mathbf{T}(s) = \frac{dC}{dz}\frac{dz}{ds} = \frac{dC}{dz}\frac{1}{L} = \mathbf{T}(Lz)$ and fixing an origin $(r_x^O; r_y^O)$ one can deduce that

$$C(z) = L\int_0^z \mathbf{T}(L\tilde{z})d\tilde{z} + (r_x^O; r_y^O) = L\int_0^z e^{i\{\int_0^{\tilde{z}} \hat{\kappa}(\bar{z})d\bar{z}+\varphi_0\}}d\tilde{z} + (r_x^O; r_y^O), \tag{180}$$

which can be trivially discretized as follows

$$C(z_k) = C(\frac{k}{N}) = \frac{L}{N}\sum_{l=0}^{k} e^{\{\frac{i}{N}\sum_{j=0}^{l}\hat{\kappa}(j)+i\varphi_0\}} + r_x^O + ir_y^O. \tag{181}$$

Hence, it is possible to reconstruct a curve using a curvature field defined for a parametrization on the unit circle. This reconstruction can be made unique by fixing the three rigid transformation parameters φ_0, r_x^O and r_y^O. The reconstruction method for possibly non-convex curves alluded to above appears as a generalized version of the Gauss-Minkowski reconstruction scheme for convex curves.

Figure 58 presents a non-convex (star-shaped) curve reconstructed from its scaled curvature. The scaled curvature is encoded via particles as presented above. The number of particles can be visualized in the figure on the graph representing the number of particles λ minus the number of particles μ for each site of T_N.

7.6 Experimental results, stochastic curvature evolution

Figure 59, presents the stochastic curvature evolution of a exemplar non-convex curve. Equation (153) *only* was simulated, using the methodology described in the section

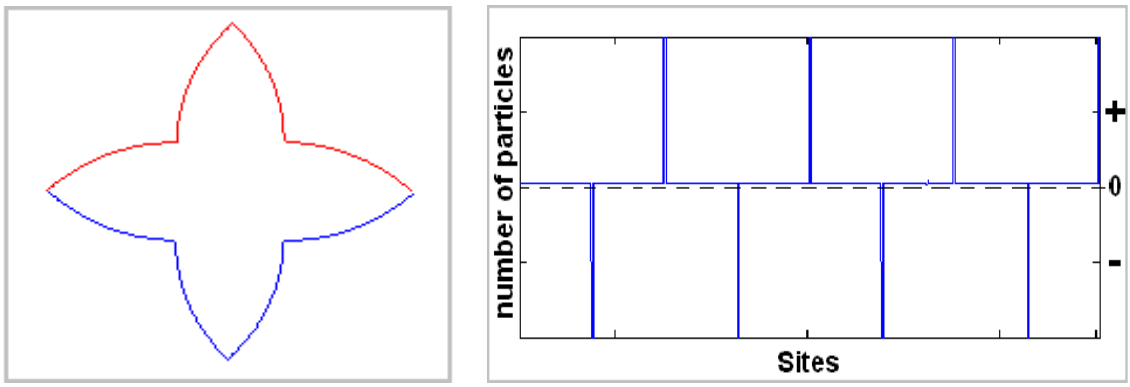


Figure 57: 4-branches star. *left:* Initial curve; *Right:* Corresponding particles encoding the scaled curvature.

above. One notes that the curve evolves towards a circle. This is in accordance with curvature flow described in [54, 55]: curves shrink to *circular* points. Note that the length does not evolve here (the curve does not shrink) since Equations (154), (155), (156) were not simulated. The graphs of particles evolves towards white noise of constant mean positive value that corresponds to a circle.

Figure 60 shows the temporal evolution of a non-convex curve. The whole system, Equations (153) to (156), was simulated. The curve shrinks to a circular point.

7.7 Chapter Conclusion

In this chapter, we proposed a methodology for the non-deterministic evolution of non-convex curves. The approach is based on the theory of interacting particles systems. Two types of particles were utilized to encode both positive and negative curvatures.

The proposed framework can be used to provide a stochastic interpretation of geodesic active contours, of interest for segmentation tasks in computer vision. Indeed, a stochastic curve evolution approach is expected to be able to escape local minima and provide better results than the deterministic approach, for the segmentation of noisy images for instance.

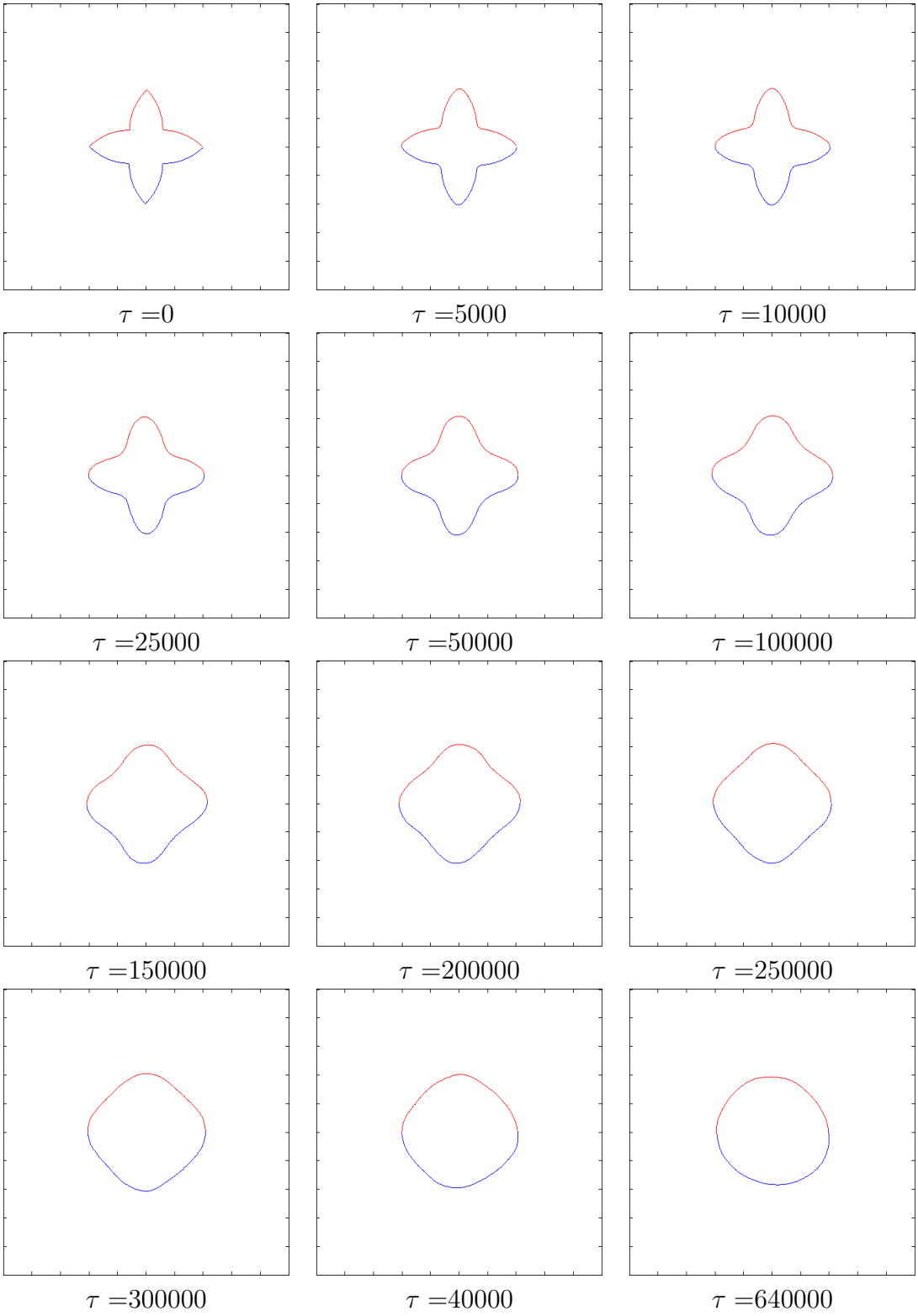


Figure 58: Stochastic curve evolution of a 4-branches star. Evolution from a star to a circle (note: the curve does not shrink to a point here. Instead it is of constant length, in accordance with our re-scaling of time)

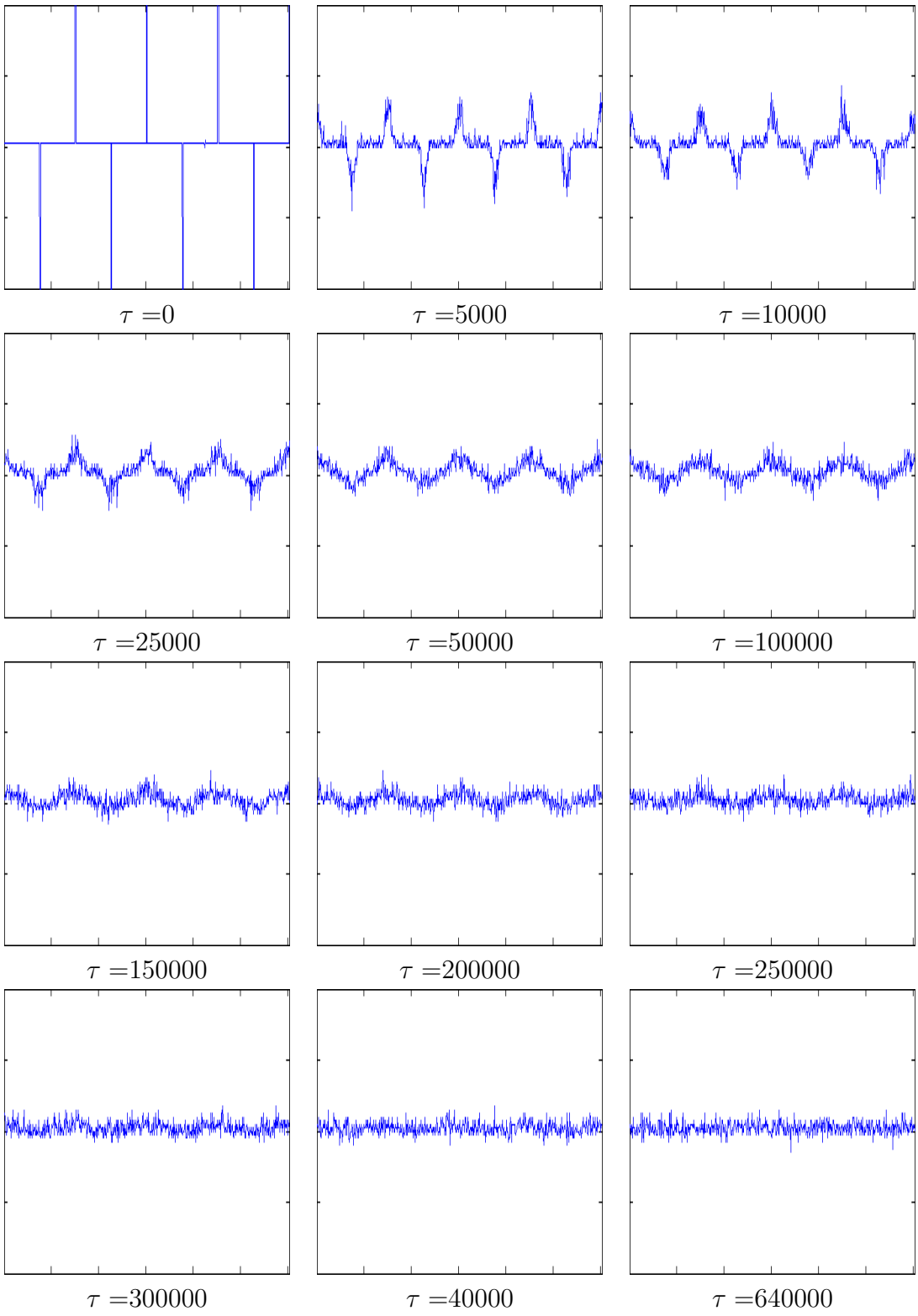


Figure 59: Stochastic (scaled) curvature evolution for a 4-branches star. Evolution of the densities of particles: From densities corresponding to a star to white noise (of non-zero mean value) corresponding to a circle.

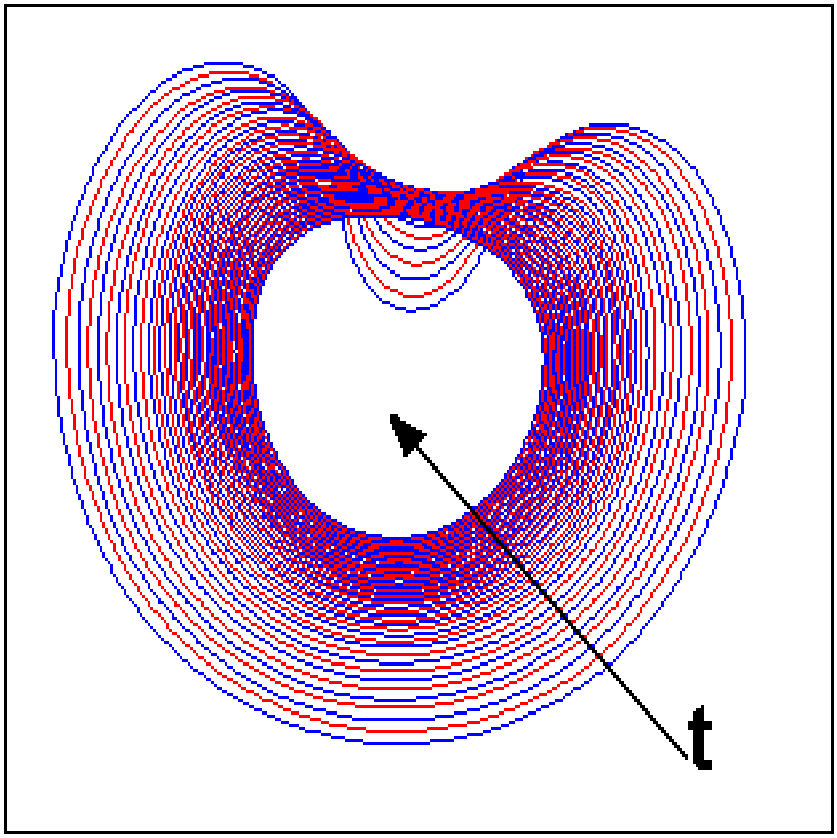


Figure 60: Stochastic curvature evolution of a non-convex curve.

The current approach has limitations that need to be overcome to be able to segment images. These limitations could be the focus of future research. For instance, the curve is currently "unaware" of where it is in the plane. A method, preferably variational, could be devised to optimize the pose of the curve in image plane based on image information. Also, no constraint in the current framework forces the curve to be closed. One could think of adding an energy term that constrains the curvatures to sum to 2π (a necessary condition for the curve to be closed) and compute the corresponding rates governing the behavior of particles. The current framework is geared towards exploiting local information. Another interesting avenue of research would be to adapt the framework to (also) perform region-based segmentation.

CHAPTER VIII

CONCLUSION

In this thesis, we proposed various computer vision techniques that can be used to perform segmentation and tracking. All the proposed techniques are based on or inspired from the geometric active contour methodology. A special emphasis was made on how one can benefit from the use of shape-driven approaches when undertaking typical computer vision tasks. The techniques proposed in this thesis combine methodologies within the field of computer vision as well as across fields such as control theory, machine learning, image processing or differential geometry, for instance. Most of the proposed techniques were cast into a variational framework that appears as a powerful and elegant approach to combine diverse methodologies.

The contributions of this thesis can be summarized as follows:

In Chapter 2, a tracking algorithm was proposed that combines a control theory technique (unscented Kalman filter) and a machine learning technique (Principal Components Analysis, PCA) to robustly track the position and the deformation of an object with active contours. One novelty of the proposed approach is that the possible shapes of the object are learned off-line and controlled during tracking.

In Chapter 3, a segmentation technique was proposed that uses image information extracted from image regions. The proposed energy functional allows one to combine an image processing technique (image thresholding) with geometric active contours. The proposed energy compares the shape of the segmenting curve with the shape extracted from the image via a dynamic thresholding operation. The technique performs segmentation based on geometric considerations of the image and contour, in

addition to purely statistical ones.

In Chapter 4, diverse machine learning methods (PCA, kernel PCA, kernel locally linear embedding...) were studied as means of including prior knowledge on shapes in the evolution process of geometric active contours. The proposed approaches enable to learn the possible shapes of objects from sets of training examples and can be used to constrain a curve to adopt familiar shapes.

In Chapter 5, a framework was proposed that combines statistical information extracted from images with shape information learned a priori from examples. The method takes advantage of the shape interpretation of the energy functionals presented in Chapters 3 and 4 to fuse image information and shape knowledge in a meaningful and homogenous manner. The resulting framework is able to robustly segment challenging images.

In Chapter 6, a technique was proposed to jointly segment a 3D object of arbitrary shape in a 2D image and estimate its 3D pose with respect to a referential attached to a unique calibrated camera. 2D segmentation with active contours and 3D registration from monocular images are two different computer vision tasks that are typically studied separately in the computer vision literature. The proposed variational approach combines both methodologies and addresses some of their typical shortcomings. The proposed method exploits the prior knowledge of the 3D shape of the object of interest.

In Chapter 7, a methodology for the non-deterministic evolution of curves was proposed that exploits the theory of interacting particles systems. Curvature driven flows affect the shape of curves and have been extensively considered from a deterministic point of view. In the chapter, a stochastic approximation of curve shortening flows was described, which provides a stochastic interpretation of geodesic active contours.

APPENDIX A

REGION BASED GRADIENTS, C-COMPUTATIONS

In this appendix we present derivations for typical region based gradients. The energies are expressed in terms of the closed regular curve C, and the corresponding gradients computed with respect to C.

A.1 General Region-Based Gradients

In this section, we compute the gradient of general region-based energies of the form

$$E(C) = \int_R f(x,y)d\Omega \tag{182}$$

where f is a given function *independent* of C, R is the region enclosed *inside* the curve C. Let $\overrightarrow{F}(x,y)$ be a vector field chosen so that $\nabla \cdot \overrightarrow{F}(x,y) = f(x,y)$. For example, given $\mathbf{x} = (x,y)$, one can choose

$$\overrightarrow{F}(x,y) = \begin{pmatrix} F^x(x,y) \\ F^y(x,y) \end{pmatrix},$$

where

$$F^x(x,y) = \frac{1}{2}\int_0^x f(\lambda,y)d\lambda,$$
$$F^y(x,y) = \frac{1}{2}\int_0^y f(x,\lambda)d\lambda,$$

so that

$$\nabla \cdot \overrightarrow{F}(x,y) = \frac{\partial F^x(x,y)}{\partial x} + \frac{\partial F^y(x,y)}{\partial y} = \frac{1}{2}f(x,y) + \frac{1}{2}f(x,y) = f(x,y). \tag{183}$$

Applying the the divergence theorem in Equation (182), one gets

$$E(C) = \int_R f(x,y)d\Omega = \oint_C \langle \overrightarrow{F}, \mathbf{N} \rangle ds, \tag{184}$$

where $\mathbf{N}$ denotes the unit normal of C and ds is the Euclidean arclength element.

To derive the gradient flow, we start by considering a fixed parametrization $p \in [0,1]$ of the curve C which does not vary as the curve evolves in time t so that (p,t) are independent variables. By a change of variable we may write E as follows:

$$E(C) = \int_0^1 \langle \overrightarrow{F}, JC_p \rangle dp, \tag{185}$$

where $J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ denotes a -90^o rotation matrix. Differentiating with respect to t yields

$$\frac{dE}{dt} = \int_0^1 \left\langle \frac{d\overrightarrow{F}}{d\mathbf{x}} C_t, JC_p \right\rangle + \left\langle \overrightarrow{F}, JC_{pt} \right\rangle dp, \tag{186}$$

where $\frac{d\overrightarrow{F}}{d\mathbf{x}}$ denotes the Jacobian matrix of $\overrightarrow{F}$ with respect to $\mathbf{x}$, that is if $\mathbf{x} = (x,y)$ and

$$\frac{d\overrightarrow{F}}{d\mathbf{x}} = \begin{bmatrix} \frac{\partial F^x}{\partial x} & \frac{\partial F^x}{\partial y} \\ \frac{\partial F^y}{\partial x} & \frac{\partial F^y}{\partial y} \end{bmatrix}. \tag{187}$$

Integrating by parts the second term of Equation (186), we obtain

$$\int_0^1 \left\langle \overrightarrow{F}, JC_{pt} \right\rangle dp = \left[\left\langle \overrightarrow{F}, JC_t \right\rangle \right]_1^0 - \int_0^1 \left\langle \frac{d\overrightarrow{F}}{d\mathbf{x}} C_p, JC_t \right\rangle dp, \tag{188}$$

$$\int_0^1 \left\langle \overrightarrow{F}, JC_{pt} \right\rangle dp = - \int_0^1 \left\langle \frac{d\overrightarrow{F}}{d\mathbf{x}} C_p, JC_t \right\rangle dp, \tag{189}$$

where we are using the fact that for a parameterized closed curve, $C(0) = C(1)$, therefore $C_t(0) = C_t(1)$. Now Equation (186) becomes

$$\frac{dE}{dt} = \int_0^1 \left\langle \frac{d\overrightarrow{F}}{d\mathbf{x}} C_t, JC_p \right\rangle - \left\langle \frac{d\overrightarrow{F}}{d\mathbf{x}} C_p, JC_t \right\rangle dp. \tag{190}$$

Rearranging terms leads to:

$$\frac{dE}{dt} = \int_0^1 \left\langle C_t, \frac{d\overrightarrow{F}}{d\mathbf{x}}^T JC_p \right\rangle - \left\langle J^T \frac{d\overrightarrow{F}}{d\mathbf{x}} C_p, C_t \right\rangle dp, \tag{191}$$

$$\frac{dE}{dt} = \int_0^1 \left\langle C_t, (\frac{d\overrightarrow{F}}{d\mathbf{x}}^T J - J^T \frac{d\overrightarrow{F}}{d\mathbf{x}}) C_p \right\rangle dp. \tag{192}$$

Since the expression

$$\frac{d\overrightarrow{F}}{d\mathbf{x}}^T J - J^T \frac{d\overrightarrow{F}}{d\mathbf{x}} = \begin{bmatrix} 0 & \frac{\partial F^x}{\partial x} + \frac{\partial F^y}{\partial y} \\ -(\frac{\partial F^x}{\partial x} + \frac{\partial F^y}{\partial y}) & 0 \end{bmatrix} = \begin{bmatrix} 0 & \nabla \cdot \overrightarrow{F} \\ -\nabla \cdot \overrightarrow{F} & 0 \end{bmatrix} = fJ,$$

Equation (192) becomes:

$$\frac{dE}{dt} = \int_0^1 \langle C_t, fJC_p \rangle \, dp = \oint_C \langle C_t . f\mathbf{N} \rangle \, ds. \tag{193}$$

We see then that the form of the gradient flow for C is revealed to be (using Cauchy-Schwartz inequality)

$$\nabla_C E = f\mathbf{N}. \tag{194}$$

Thus the flow depends only upon f, not upon the particular choice for $\overrightarrow{F}$.

One can also compute the gradient of energies defined as region integrals *outside* the curve of the form

$$E_o(C) = \int_{R^c} f(x,y) d\Omega \tag{195}$$

Noting that $\int_{R^c} f(x,y)d\Omega = \int_\Omega f(x,y)d\Omega - \int_R f(x,y)d\Omega$, and that $\int_\Omega f(x,y)d\Omega$ is independent of C, it is straightforward to see that

$$\nabla_C E_o = -f\mathbf{N}. \tag{196}$$

A.2 Particular Region based gradients

A.2.1 Average of pixel intensities inside the curve

The average of pixel intensities inside the curve can be computed as

$$\mu_{\text{in}}(C) = \frac{\int_R I(x,y) \, d\Omega}{\int_R d\Omega} = \frac{S_{\text{in}}}{A_{\text{in}}}$$

with $S_{\text{in}} = \int_R I(x,y) \, d\Omega$ and $A_{\text{in}} = \int_R d\Omega$.

From Section A.1, one has

$$\nabla_C S_{\text{in}}(x,y) = I(x,y).\mathbf{N} \qquad \text{and} \qquad \nabla_C A_{\text{in}}(x,y) = \mathbf{N}.$$

Thus, using simple rules of calculus, one gets

$$\nabla_C \mu_{\text{in}} = \frac{A_{\text{in}}.I(x,y).\mathbf{N} - S_{\text{in}}.\mathbf{N}}{A_{\text{in}}^2} = \left(\frac{I(x,y) - \mu_{\text{in}}}{A_{\text{in}}}\right)\mathbf{N}$$

A.2.2 Variance of pixel intensities inside the curve

The variance of pixel intensities inside the curve can be computed as

$$\Sigma_{\text{in}}(C) = \frac{\int_R \big(I(x,y) - \mu_{\text{in}}\big)^2 d\Omega}{\int_R d\Omega} = \frac{U_{\text{in}}}{A_{\text{in}}}$$

with $U_{\text{in}} = \int_R \big(I(x,y) - \mu_{\text{in}}\big)^2 d\Omega$ and $A_{\text{in}} = \int_R d\Omega$.

From Section A.1, one has

$$\nabla_C U_{\text{in}}(x,y) = \big(I(x,y) - \mu_{\text{in}}\big)^2.\mathbf{N} \qquad \text{and} \qquad \nabla_C A_{\text{in}}(x,y) = \mathbf{N}.$$

Thus, using simple rules of calculus, one gets

$$\nabla_C \Sigma_{\text{in}} = \frac{A_{\text{in}}.\big(I(x,y) - \mu_{\text{in}}\big)^2.\mathbf{N} - U_{\text{in}}.\mathbf{N}}{A_{\text{in}}^2} = \left(\frac{\big(I(x,y) - \mu_{\text{in}}\big)^2 - \Sigma_{\text{in}}}{A_{\text{in}}}\right).\mathbf{N}$$

A.2.3 Average and Variance of pixel intensities outside the curve

Using the gradient computations for energies defined on regions outside the curve in Section A.1 and the results above for $\nabla_C \mu_{\text{in}}$ and $\nabla_C \Sigma_{\text{in}}$, one immediately gets

$$\nabla_C \mu_{\text{out}} = -\left(\frac{I(x,y) - \mu_{\text{out}}}{A_{\text{out}}}\right)\mathbf{N} \qquad \text{and}$$

$$\nabla_C \Sigma_{\text{out}} = -\left(\frac{\big(I(x,y) - \mu_{\text{out}}\big)^2 - \Sigma_{\text{out}}}{A_{\text{out}}}\right).\mathbf{N}$$

A.2.4 Chan and Vese

The region-based energy proposed by Chan and Vese [12] segment regions with different averages of pixel intensities. The energy is

$$E_{\text{CV}} = \int_R (I(x,y) - \mu_{\text{in}}(C))^2 d\Omega + \int_{R^c} (I(x,y) - \mu_{\text{out}}(C))^2 d\Omega \tag{197}$$

Note that, here, the functions integrated on the different regions, namely $(I(x,y) - \mu_{\text{in}}(C))^2$ and $(I(x,y) - \mu_{\text{out}}(C))^2$, are *dependent* on C. Differentiating with respect to artificial time t, one gets

$$\begin{aligned}\frac{dE_{\text{CV}}}{dt} = &\oint_C \left\langle (I(x,y) - \mu_{\text{in}})^2 . \mathbf{N}, C_t \right\rangle ds - \oint_C \left\langle (I(x,y) - \mu_{\text{out}})^2 . \mathbf{N}, C_t \right\rangle ds \\ &- \int_R 2(I(x,y) - \mu_{\text{in}}) \left\langle \nabla_C \mu_{\text{in}}, C_t \right\rangle d\Omega \\ &- \int_{R^c} 2(I(x,y) - \mu_{\text{out}}) \left\langle \nabla_C \mu_{\text{out}}, C_t \right\rangle d\Omega\end{aligned} \tag{198}$$

Using the fact that the terms $\langle \nabla_C \mu_{\text{out}}, C_t \rangle$ and $\langle \nabla_C \mu_{\text{in}}, C_t \rangle$ are independent of $\mathbf{x} = (x,y)$, one gets

$$\begin{aligned}\frac{dE_{\text{CV}}}{dt} &= \oint_C \left\langle \big((I(x,y) - \mu_{\text{in}})^2 - (I(x,y) - \mu_{\text{out}})^2\big) . \mathbf{N}, C_t \right\rangle ds \\ &\quad - 2 \left\langle \nabla_C \mu_{\text{in}}, C_t \right\rangle . \int_R (I(x,y) - \mu_{\text{in}}) d\Omega \\ &\quad - 2 \left\langle \nabla_C \mu_{\text{out}}, C_t \right\rangle . \int_{R^c} (I(x,y) - \mu_{\text{out}}) d\Omega \\ &= \oint_C \left\langle \big((I(x,y) - \mu_{\text{in}})^2 - (I(x,y) - \mu_{\text{out}})^2\big) . \mathbf{N}, C_t \right\rangle ds \\ &\quad - 2 \left\langle \nabla_C \mu_{\text{in}}, C_t \right\rangle . \underbrace{(A_{\text{in}}\mu_{\text{in}} - A_{\text{in}}\mu_{\text{in}})}_{=0} \\ &\quad - 2 \left\langle \nabla_C \mu_{\text{out}}, C_t \right\rangle . \underbrace{(A_{\text{out}}\mu_{\text{out}} - A_{\text{out}}\mu_{\text{out}})}_{=0} \\ &= \oint_C \left\langle \big((I(x,y) - \mu_{\text{in}})^2 - (I(x,y) - \mu_{\text{out}})^2\big) . \mathbf{N}, C_t \right\rangle ds\end{aligned} \tag{199}$$

From where, one deduces (using Cauchy-Schwartz inequality) that

$$\nabla_C E_{\text{CV}} = \big((I(x,y) - \mu_{\text{in}})^2 - (I(x,y) - \mu_{\text{out}})^2\big) . \mathbf{N}$$

A.2.5 Paragios and Deriche

The region-based energy proposed by Paragios and Deriche [93, 106] segment regions with different Gaussian distributions of pixel intensities. The energy is

$$\begin{aligned} E_{\mathrm{PD}} = & \int_R \left(\log(\Sigma_{\mathrm{in}}(C)) + \frac{(I(x,y) - \mu_{\mathrm{in}}(C))^2}{\Sigma_{\mathrm{in}}(C)} \right) d\Omega \\ & + \int_{R^c} \left(\log(\Sigma_{\mathrm{out}}(C)) + \frac{(I(x,y) - \mu_{\mathrm{out}}(C))^2}{\Sigma_{\mathrm{out}}(C)} \right) d\Omega \end{aligned} \tag{200}$$

Note that, here again, the functions integrated on the different regions are *dependent* on C. Differentiating with respect to artificial time t, one gets

$$\begin{aligned} \frac{dE_{\mathrm{PD}}}{dt} = & \oint_C \left\langle \left(\log(\Sigma_{\mathrm{in}}) + \frac{(I(x,y) - \mu_{\mathrm{in}})^2}{\Sigma_{\mathrm{in}}} \right) .\mathbf{N}, C_t \right\rangle ds \\ & - \oint_C \left\langle \left(\log(\Sigma_{\mathrm{out}}) + \frac{(I(x,y) - \mu_{\mathrm{out}})^2}{\Sigma_{\mathrm{out}}} \right) .\mathbf{N}, C_t \right\rangle ds \\ & - \int_R 2 \frac{(I(x,y) - \mu_{\mathrm{in}})}{\Sigma_{\mathrm{in}}} \langle \nabla_C \mu_{\mathrm{in}}, C_t \rangle \, d\Omega - \int_{R^c} 2 \frac{(I(x,y) - \mu_{\mathrm{out}})}{\Sigma_{\mathrm{out}}} \langle \nabla_C \mu_{\mathrm{out}}, C_t \rangle \, d\Omega \\ & + \int_R \left(\frac{1}{\Sigma_{\mathrm{in}}} - \frac{(I(x,y) - \mu_{\mathrm{in}})^2}{\Sigma_{\mathrm{in}}^2} \right) . \langle \nabla_C \Sigma_{\mathrm{in}}, C_t \rangle \, d\Omega \\ & + \int_{R^c} \left(\frac{1}{\Sigma_{\mathrm{out}}} - \frac{(I(x,y) - \mu_{\mathrm{out}})^2}{\Sigma_{\mathrm{out}}^2} \right) . \langle \nabla_C \Sigma_{\mathrm{out}}, C_t \rangle \, d\Omega \end{aligned} \tag{201}$$

Using the fact that the terms $\langle \nabla_C \mu_{\mathrm{out}}, C_t \rangle$, $\langle \nabla_C \mu_{\mathrm{in}}, C_t \rangle$, $\langle \nabla_C \Sigma_{\mathrm{in}}, C_t \rangle$ and $\langle \nabla_C \Sigma_{\mathrm{out}}, C_t \rangle$ are independent of $\mathbf{x} = (x, y)$, one gets

$$\begin{aligned} \frac{dE_{\mathrm{PD}}}{dt} = & \oint_C \left\langle \left(\log \left(\frac{\Sigma_{\mathrm{in}}}{\Sigma_{\mathrm{out}}} \right) + \frac{(I(x,y) - \mu_{\mathrm{in}})^2}{\Sigma_{\mathrm{in}}} \frac{(I(x,y) - \mu_{\mathrm{out}})^2}{\Sigma_{\mathrm{out}}} \right) .\mathbf{N}, C_t \right\rangle ds \\ & - 2 \frac{\langle \nabla_C \mu_{\mathrm{in}}, C_t \rangle}{\Sigma_{\mathrm{in}}} \underbrace{\int_R (I(x,y) - \mu_{\mathrm{in}}) d\Omega}_{=0} - 2 \frac{\langle \nabla_C \mu_{\mathrm{out}}, C_t \rangle}{\Sigma_{\mathrm{out}}} \underbrace{\int_{R^c} (I(x,y) - \mu_{\mathrm{out}}) d\Omega}_{=0} \\ & + \frac{\langle \nabla_C \Sigma_{\mathrm{in}}, C_t \rangle}{\Sigma_{\mathrm{in}}^2} \underbrace{\int_R \left(\Sigma_{\mathrm{in}} - (I(x,y) - \mu_{\mathrm{in}})^2 \right) d\Omega}_{A_{\mathrm{in}} \Sigma_{\mathrm{in}} - A_{\mathrm{in}} \Sigma_{\mathrm{in}} = 0} \\ & + \frac{\langle \nabla_C \Sigma_{\mathrm{out}}, C_t \rangle}{\Sigma_{\mathrm{out}}^2} \underbrace{\int_{R^c} \left(\Sigma_{\mathrm{out}} - (I(x,y) - \mu_{\mathrm{out}})^2 \right) d\Omega}_{A_{\mathrm{out}} \Sigma_{\mathrm{out}} - A_{\mathrm{out}} \Sigma_{\mathrm{out}} = 0} \end{aligned} \tag{202}$$

From where, one deduces (using Cauchy-Schwartz inequality) that

$$\nabla_C E_{\mathrm{PD}} = \left(\log \left(\frac{\Sigma_{\mathrm{in}}}{\Sigma_{\mathrm{out}}} \right) + \frac{(I(x,y) - \mu_{\mathrm{in}})^2}{\Sigma_{\mathrm{in}}} - \frac{(I(x,y) - \mu_{\mathrm{out}})^2}{\Sigma_{\mathrm{out}}} \right) .\mathbf{N}$$

N.B.: Hence, from the computations above, one notices that both the Chan-Vese and Paragios-Deriche flows are "blind" to changes in statistics (inside and outside C) that are due to the deformation of the curve. This is in contrast with the flows presented in Chapter 3, Sections 3.4.1.1 and 3.4.1.2, which take the variation of regions statistics into account leading to better segmentation results.

APPENDIX B

REGION BASED GRADIENTS, ϕ-COMPUTATIONS

In this appendix we detail derivations for region based gradients of energies. The energies are expressed in terms of the signed distance function ϕ (representing a curve C), and the corresponding gradients computed with respect to ϕ. The Heaviside and Dirac delta functions are defined in the distributional sense.

B.1 General Region-Based Gradients

In this section, we compute the gradient of general region-based energies of the form

$$E(\phi) = \int_R f(x,y)d\Omega = \int_\Omega H\phi(x,y).f(x,y)d\Omega \tag{203}$$

where f is a given function *independent* of C, R is the region enclosed *inside* the curve C. Let ν be a perturbation of *phi* (of same dimension as ϕ) and ξ a real number. One can write

$$E(\phi + \xi\nu) = \int_\Omega H(\phi(x,y) + \xi\nu(x,y)).f(x,y)d\Omega \tag{204}$$

Taking the Gâteaux derivative [47] of $E(\phi + \xi\nu)$ with respect to ξ, one gets

$$\frac{dE(\phi + \xi\nu)}{d\xi}_{|\xi=0} = \int_\Omega \delta(\phi(x,y)).\nu(x,y).f(x,y)d\Omega \tag{205}$$

From where, one deduces (using Cauchy-Schwartz inequality) that

$$\nabla_\phi E = \delta(\phi).f \tag{206}$$

N.B.1: Comparing to Appendix A, Section A.1, one can realize how easily the result above was derived. This simplicity of computation is one of the main reasons

why methods involving the definition of region energies directly in terms of the SDF ϕ (instead of C) enjoyed a enormous success in the field. This type of approach was introduced in [12].

N.B.2: Comparing to Appendix A, Section A.1, one can realize that $\delta(\phi)$ plays the role for ϕ-formulation of $\mathbf{N}$ for C-formulation. The term $\delta(\phi)$ also moves the zero level set "normally".

B.2 Averages and Variances of regions

From Equations (81) and (82), it is straightforward to see that

$$\begin{aligned}
\frac{\partial \mu_{\text{in}}(\phi+\xi\nu)}{\partial \xi}|_{\xi=0} &= \frac{\int_\Omega \delta\phi (I-\mu_{\text{in}}(\phi))\nu d\Omega}{A_{\text{in}}} \\
\frac{\partial \Sigma_{\text{in}}(\phi+\xi\nu)}{\partial \xi}|_{\xi=0} &= \frac{1}{A_{\text{in}}}\int_\Omega \delta\phi\big((I-\mu_{\text{in}}(\phi))(I-\mu_{\text{in}}(\phi))^T-\Sigma_{\text{in}}(\phi)\big)\nu d\Omega
\end{aligned} \tag{207}$$

Using Cauchy-Schwartz inequality, one can show that $\frac{\partial \mu_{\text{in}}(\phi+\xi\nu)}{\partial \xi}|_{\xi=0}$ is maximum for

$$\nu_0 = \delta\phi.\left(\frac{I-\mu_{\text{in}}(\phi)}{A_{\text{in}}}\right)$$

Similarly, $\frac{\partial \Sigma_{\text{in}}(\phi+\xi\nu)}{\partial \xi}|_{\xi=0}$ is maximum for

$$\nu_0 = \delta\phi.\left(\frac{(I-\mu_{\text{in}}(\phi))(I-\mu_{\text{in}}(\phi))^T-\Sigma_{\text{in}}(\phi)}{A_{\text{in}}}\right).$$

By definition of the gradient, this proves Equation (85).

B.3 TAC: Implicit thresholds - Gaussian case

We now detail the derivation of $\nabla_\phi E_{\text{image}}$ in Equation (83). The functional G as defined in Equation (46) depends on ϕ, $G(x) = G[\phi](x)$. Let us compute the first variation of $E_{\text{image}}(\phi)$ as defined in Equation (29). Let $\xi \in \mathbf{R}$ and $\nu \in \mathcal{T}_M$ ($\mathcal{T}_M$ being the tangential space to the space of signed distance functions).

$$\frac{\partial E(\phi+\xi\nu)}{\partial \xi}|_{\xi=0} = \int_\Omega \nu.[H\phi - G](\delta\phi)d\Omega - \int_\Omega [H\phi-G].\frac{\partial G[\phi+\xi\nu]}{\partial \xi}|_{\xi=0}d\Omega. \tag{208}$$

Using Equation (46), we can write

$$\frac{\partial G[\phi+\xi\nu]}{\partial\xi}|_{\xi=0} = \delta_{\epsilon_2}\left(\log\left(\frac{P_{\text{in}}}{P_{\text{out}}}\right)\right).\Big[\frac{\partial\log(P_{\text{in}}(\phi+\xi\nu))}{\partial\xi} - \frac{\partial\log(P_{\text{out}}(\phi+\xi\nu))}{\partial\xi}\Big]|_{\xi=0} \tag{209}$$

From Equation (80), we get

$$\log(P_{\text{in}}) = -\frac{1}{2}\log(|\Sigma_{\text{in}}|) - \frac{1}{2}(I-\mu_{\text{in}})^T\Sigma_{\text{in}}^{-1}(I-\mu_{\text{in}}) \tag{210}$$

Using the chain rule, we can write

$$\frac{\partial\log(P_{\text{in}}(\phi+\xi\nu))}{\partial\xi}|_{\xi=0} = \frac{\partial\log(P_{\text{in}})}{\partial\mu_{\text{in}}}\frac{\partial\mu_{\text{in}}(\phi+\xi\nu)}{\partial\xi}|_{\xi=0} + \frac{\partial\log(P_{\text{in}})}{\partial\Sigma_{\text{in}}}\frac{\partial\Sigma_{\text{in}}(\phi+\xi\nu)}{\partial\xi}|_{\xi=0} \tag{211}$$

The gradients of $\log(P_{\text{in}})$ with respect to μ_{in} and Σ_{in} can be easily computed from Equation (210) as

$$\frac{\partial\log(P_{\text{in}})}{\partial\mu_{\text{in}}} = \Sigma_{\text{in}}^{-1}(I-\mu_{\text{in}})\frac{\partial\log(P_{\text{in}})}{\partial\Sigma_{\text{in}}} = \frac{1}{2}\big[\Sigma_{\text{in}}^{-1}(I-\mu_{\text{in}})(I-\mu_{\text{in}})^T\Sigma_{\text{in}}^{-1} - \Sigma_{\text{in}}^{-1}\big]. \tag{212}$$

In the case of color images, $\frac{\partial\log(P_{\text{in}})}{\partial\mu_{\text{in}}}$ and $\frac{\partial\log(P_{\text{in}})}{\partial\Sigma_{\text{in}}}$ are taken as row vectors respectively in $\mathbf{R}^3$ and $\mathbf{R}^9$. Equation (207) can be rewritten more compactly as

$$\begin{aligned}\frac{\partial\mu_{\text{in}}(\phi+\xi\nu)}{\partial\xi}|_{\xi=0} &= \int_\Omega \nabla_\phi\mu_{\text{in}}(x).\nu(x)\ d\Omega \\ \frac{\partial\Sigma_{\text{in}}(\phi+\xi\nu)}{\partial\xi}|_{\xi=0} &= \int_\Omega \nabla_\phi\Sigma_{\text{in}}(x).\nu(x)\ d\Omega\end{aligned} \tag{213}$$

Equations (211), (212) and (213) can be computed for P_{out} by replacing μ_{in} and Σ_{in} by μ_{out} and Σ_{out}, respectively. Using the results concerning the statistics inside the curve of Equations (213) and (212), as well as their counterparts involving the statistics outside C, and successively plugging into Equations (211), (209) and (208), one can write

$$\frac{\partial E(\phi+\xi\nu)}{\partial\xi}|_{\xi=0} = \int_\Omega \nu.\big[\delta\phi(H\phi-G) + \beta_\mu^{\text{out}}\nabla_\phi\mu_{\text{out}} + \beta_\Sigma^{\text{out}}\nabla_\phi\Sigma_{\text{out}} - \beta_\mu^{\text{in}}\nabla_\phi\mu_{\text{in}} - \beta_\Sigma^{\text{in}}\nabla_\phi\Sigma_{\text{in}}\big]\ d\Omega. \tag{214}$$

where the coefficients β_μ^{in}, β_Σ^{in}, β_μ^{out} and $\beta_\Sigma^{\text{out}}$ are independent of the integration variable x and are defined in Equation (84). Using Cauchy-Schwartz inequality, one finds that $\frac{\partial E(\phi+\xi\nu)}{\partial \xi}|_{\xi=0}$ is maximum for $\nu_0 = \delta\phi(H\phi - G) + \beta_\mu^{\text{out}}.\nabla_\phi \mu_{\text{out}} + \beta_\Sigma^{\text{out}}.\nabla_\phi \Sigma_{\text{out}} - \beta_\mu^{\text{in}}.\nabla_\phi \mu_{\text{in}} - \beta_\Sigma^{\text{in}}.\nabla_\phi \Sigma_{\text{in}}$, which proves Equation (83) by definition of the gradient.

B.4 TAC: Implicit thresholds - Generalized distribution case

In what follows, we will use global intensity distributions to threshold the image. In contrast, with Equation (46), we define G as (note, no log)

$G[x, y] = G\big[\big(P_{\text{in}} - P_{\text{out}}\big)\big(I(x)\big)\big]$[1]. In this expression, the likelihoods P_{in} and P_{out} represent the knowledge of the statistics of the object and background, respectively. These likelihoods are computed from the segmenting contour as

$$P_{\text{in}}(z, \phi) = \frac{\int_\Omega \mathbf{K}(I(u) - z) H\phi(u) du}{\int_\Omega H\phi(u) du} = \frac{\int_\Omega \mathbf{K}(I(u) - z) H\phi(u) du}{A_{\text{in}}} \tag{215}$$

and

$$P_{\text{out}}(z, \phi) = \frac{\int_\Omega \mathbf{K}(I(u) - z)\{1 - H\phi(u)\} du}{\int_\Omega \{1 - H\phi(u)\} du} = \frac{\int_\Omega \mathbf{K}(I(u) - z)\{1 - H\phi(u)\} du}{A_{\text{out}}} \tag{216}$$

with $\mathbf{K}(\chi)$ typically being a smooth version of the Dirac function, e.g.:

$$\mathbf{K}(\chi) = \frac{1}{\sqrt{2\pi\sigma^2}}.e^{\frac{\chi^2}{2\sigma^2}}$$

for a chosen small value of σ.

Let us compute the first derivative of $P_{\text{in}}(z, \phi + \xi\nu)$ and $P_{\text{out}}(z, \phi + \xi\nu)$, with respect to ξ

$$\frac{dP_{\text{in}}(z, \phi + \xi\nu)}{d\xi}_{|\xi=0} = \frac{\int_\Omega [\mathbf{K}(I(x) - z) - P_{\text{in}}(z)] \delta\phi(x)\nu(x) dx}{A_{\text{in}}} \tag{217}$$

[1]Here, $s = \big(P_{\text{in}} - P_{\text{out}}\big)\big(I(x)\big)$, the signed distance between global statistics inside and outside the contour. If $P_{\text{in}}\big(I(x)\big) > P_{\text{out}}\big(I(x)\big)$, $G[s(x)]$ is close to 1; otherwise $G[s(x)]$ is close to 0.

$$\frac{dP_{\text{out}}(z,\phi+\xi\nu)}{d\xi}_{|\xi=0} = \frac{\int_\Omega [P_{\text{out}}(z) - \mathbf{K}(I(x)-z)]\delta\phi(x)\nu(x)dx}{A_{\text{out}}} \tag{218}$$

In what follows, we write $P_{\text{in}}(z,\phi) = P_{\text{in}}(z)$ and $P_{\text{out}}(z,\phi) = P_{\text{out}}(z)$, to simplify notation. Let us now consider the first derivative of $E_{\text{image}}(\phi+\xi\nu)$, with respect to $\xi \in \mathbf{R}^+$ in the direction ν.

$$\begin{aligned}
&\frac{dE_{\text{image}}(\phi+\xi\nu)}{d\xi}_{|\xi=0} \\
&= 2\int_\Omega \{H\phi(x) - G\big[\big(P_{\text{in}} - P_{\text{out}}\big)\big(I(x)\big)\big]\}. \\
&\{\nu(x)\delta\phi(x) - G'\big[\big(P_{\text{in}} - P_{\text{out}}\big)\big(I(x)\big)\big].\frac{d\big[P_{\text{in}} - P_{\text{out}}\big](I(x),\phi+\xi\nu)}{d\xi}_{|\xi=0}\}dx \\
&= 2\int_\Omega \{H\phi(x) - G\big[\big(P_{\text{in}} - P_{\text{out}}\big)\big(I(x)\big)\big]\}.\delta\phi(x).\nu(x)dx + A - B
\end{aligned} \tag{219}$$

with A and B computed using Equations (217) and (218), as follows

$$\begin{aligned}
A = 2\int_\Omega^{u,v} \delta\phi(u)\nu(u)\int_\Omega^{x,y} \{H\phi(x) - G\big[\big(P_{\text{in}} - P_{\text{out}}\big)\big(I(x)\big)\big]\}\times \\
\big[\frac{P_{\text{in}}(I(x))}{A_{\text{in}}} + \frac{P_{\text{out}}(I(x))}{A_{\text{out}}}\big]G'\big[\big(P_{\text{in}} - P_{\text{out}}\big)\big(I(x)\big)\big]dx.du
\end{aligned} \tag{220}$$

$$\begin{aligned}
B = 2\int_\Omega^{u,v} \delta\phi(u)\nu(u)\int_\Omega^{x,y} \{H\phi(x) - G\big[\big(P_{\text{in}} - P_{\text{out}}\big)\big(I(x)\big)\big]\}\times \\
\mathbf{K}(I(u)-I(x))\big[\frac{1}{A_{\text{in}}} + \frac{1}{A_{\text{out}}}\big]G'\big[\big(P_{\text{in}} - P_{\text{out}}\big)\big(I(x)\big)\big]dx.du
\end{aligned} \tag{221}$$

In Equation (220), the term

$$a = \int_\Omega^{x,y} \{H\phi(x) - G\big[\big(P_{\text{in}} - P_{\text{out}}\big)\big(I(x)\big)\big]\}.\big[\frac{P_{\text{in}}(I(x))}{A_{\text{in}}} + \frac{P_{\text{out}}(I(x))}{A_{\text{out}}}\big]G'\big[\big(P_{\text{in}} - P_{\text{out}}\big)\big(I(x)\big)\big]dx$$

is independent of u, and Equation (220) can be rewritten as

$$A = 2\int_\Omega^{u,v} a.\delta\phi(u).\nu(u)du \tag{222}$$

In Equation (221), one can define

$$b(u) = \int_{\Omega}^{x,y} \{H\phi(x) - G\big[\big(P_{\text{in}} - P_{\text{out}}\big)\big(I(x)\big)\big]\}\times \\ \mathbf{K}(I(u) - I(x))\big[\frac{1}{A_{\text{in}}} + \frac{1}{A_{\text{out}}}\big]G'\big[\big(P_{\text{in}} - P_{\text{out}}\big)\big(I(x)\big)\big]dx \tag{223}$$

The term B then becomes

$$B = 2\int_{\Omega}^{u,v} b(u).\delta\phi(u).\nu(u)du. \tag{224}$$

Using the a Dirac impulse function for the Kernel $\mathbf{K}$, i.e. $\mathbf{K}(\chi) = \delta(\chi)$, one can note that $\mathbf{K}(I(u) - I(x)) \neq 0$ for $I(u) = I(x)$. Thus, Equation (223) can be written as follows

$$b(u) = \big[\frac{1}{A_{\text{in}}} + \frac{1}{A_{\text{out}}}\big].G'\big[\big(P_{\text{in}} - P_{\text{out}}\big)\big(I(u)\big)\big] \\ \times \int_{\Omega}^{x,y} \{H\phi(x) - G\big[\big(P_{\text{in}} - P_{\text{out}}\big)\big(I(u)\big)\big]\}\mathbf{K}(I(u) - I(x))dx$$

$$b(u) = \big[\frac{1}{A_{\text{in}}} + \frac{1}{A_{\text{out}}}\big].G'\big[\big(P_{\text{in}} - P_{\text{out}}\big)\big(I(u)\big)\big] \\ \times \{P_{\text{in}}\big(I(u)\big).A_{\text{in}} - G\big[\big(P_{\text{in}} - P_{\text{out}}\big)\big(I(u)\big)\big].\big[P_{\text{in}}\big(I(u)\big).A_{\text{in}} + P_{\text{out}}\big(I(u)\big).A_{\text{out}}\big]\} \tag{225}$$

Using Equations (219), (222) and (224), one gets

$$\frac{dE_{\text{image}}(\phi + \xi\nu)}{d\xi}_{|\xi=0} = 2\int_{\Omega} \nu(u).\{H\phi(u) - G\big[\big(P_{\text{in}} - P_{\text{out}}\big)\big(I(u)\big)\big] + a - b(u)\}\delta\phi(u)du \tag{226}$$

By the Cauchy-Schwartz inequality, the variation of $E_{\text{image}}(\phi)$ is maximum in the direction

$$\nu_0(x) = \{H\phi(u) - G\big[\big(P_{\text{in}} - P_{\text{out}}\big)\big(I(u)\big)\big] + a - b(u)\}\delta\phi(u) \tag{227}$$

Thus, $\nabla_{\phi}E_{\text{image}} = \alpha\nu_0$, with $\alpha \in \mathbf{R}^+$, by definition of the gradient.

REFERENCES

[1] Adalsteinsson, D. and Sethian, J. A., "A fast level set method for propagating interfaces," *Journal of Computational Physics*, vol. 118, pp. 269–277, 1995.

[2] Angenent, S., Pichon, E., , and Tannenbaum, A., "Mathematical methods in medical image processing," *Bulletin of the American Mathematical Society*, 2005.

[3] Angenent, S., Tannenbaum, A., Yezzi, A., and Zeitouni, O., "Curve shortening and interacting particle systems," *to appear in volume edited by H. Krim and A. Yezzi.*

[4] Arias, P., Randall, G., and Sapiro, G., "Connecting the out-of-sample and pre-image problems in kernel methods," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2007.

[5] Arous, G. B., Tannenbaum, A., and Zeitouni, O., "Stochastic approximations of curve shortening flows," *Journal of Differential Equations*, vol. 195, 2003.

[6] Ayache, N., "Medical computer vision, virtual reality and robotics," *Image and Vision Computing*, vol. 13, no. 4, pp. 295–313, 1995.

[7] Bengio, Y., Paiement, J., and Vincent, P., "Out-of-sample extensions for lle, isomaps, mds, eigenmaps and spectral clustering," in *Advances in NIPS*, vol. 16, 2004.

[8] Blake, A. and Isard, M., eds., *Active Contours*. Springer, 1998.

[9] Caselles, V., Catte, F., Coll, T., and Dibos, F., "A geometric model for active contours in image processing," *Numerische Mathematik*, vol. 66, pp. 1–31, 1993.

[10] Caselles, V., Kimmel, R., and Sapiro, G., "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.

[11] Chan, T., Sandberg, B., and Vese, L., "Active contours without edges for vector-valued images," *Journal of Visual Communication and Image Representation*, vol. 11, pp. 130–141, 2000.

[12] Chan, T. and Vese, L., "Active contours without edges," *IEEE Trans. on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[13] Chan, T. and Zhu, W., "Level set based shape prior segmentation," tech. rep., Computational Applied Mathematics, UCLA, 2003.

[14] Chan, T. and Zhu, W., "Level set based shape prior segmentation," in *Proc. CVPR*, vol. 2, pp. 1164–1170, IEEE, 2005.

[15] Chen, Y., Huang, T., and Rui, Y., "Parametric contour tracking using unscented kalman filter," *Proceedings of the International Conference on Image Processing*, vol. 3, no. 3, pp. 613–616, 2002.

[16] Chen, Y., Thiruvenkadam, S., and Huang, F., "On the incorporation of shape priors into geometric active contours," in *vlsm*, p. 145, Feb.

[17] Chopp., D. L., "Computing minimal surfaces via level set curvature flow.," *Journal of Computational Physics*, vol. 106, pp. 77–91, 1993.

[18] Cootes, T., Beeston, C., G.Edwards, and Taylor, C., "Unified framework for atlas matching using active appearance models," in *Int'l Conf. Information Processing in Med. Imaging*, pp. 322–333, Springer-Verlag, 1999.

[19] Cootes, T., Taylor, C., and Cooper, D., "Active shape models-their training and application," in *Comput.Vis. Image Understanding*, vol. 61, pp. 38–59, 1995.

[20] Cremers, D., Kohlberger, T., and Schnoerr, C., "Diffusion snakes: introducing statistical shape knowledge into the Mumford-Shah functional," in *International journal of computer vision*, vol. 50, pp. 295–313, Feb. 2002.

[21] Cremers, D., Kohlberger, T., and Schnoerr, C., "Shape statistics in kernel space for variational image segmentation," in *Pattern Recognition*, vol. 36, pp. 1292–1943, 2003.

[22] Cremers, D., Osher, S., and Soatto., S., "Kernel density estimation and intrinsic alignment for knowledge-driven segmentation: teaching level sets to walk," in *Proc. of DAGM*, vol. 3157, pp. 36–44, 2004.

[23] Cremers, D., Sochen, N., and Schnoerr, C., "A multiphase dynamic labeling model for variational recognition-driven image segmentation," *Intl Journal of Computer Vision*, 2005.

[24] Dambreville, S., Rathi, Y., and Tannenbaum., A., "Non linear shape prior from kernel space for geometric active contours," in *Proceedings of SPIE*, vol. 6064, pp. 404–412, 2006.

[25] Dambreville, S., Rathi, Y., and Tannenbaum, A., "A shape-based approach to robust image segmentation," in *Int. Conf. on Image Analysis and Recognotion*, vol. 1, pp. 173–183, 2006.

[26] Dambreville, S., Rathi, Y., and Tannenbaum., A., "Shape-based approach to robust image segmentation using kernel pca," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 977–984, 2006.

[27] Dambreville, S., Rathi, Y., and Tannenbaum, A., "Unscented kalman filtering applied to geometric active contours for tracking deformable objects," in *American Control Conference*, 2006.

[28] Dambreville, S., Rathi, Y., and Tannenbaum, A., "A framework for image segmentation using shape models and kernel space shape priors," *Trans. Pattern Analysis and Machine Intelligence*, to appear.

[29] Dambreville, S., Yezzi, A., Niethammer, M., and Tannenbaum, A., "A variational framework combining level-sets and thresholding.," in *British Machine Vision Conference (BMVC)*, pp. 266–280, 2007.

[30] Dambreville, S., Yezzi, A., Niethammer, M., and Tannenbaum, A., "A variational segmentation framework using active contours and thresholding.," in *Int. Conf. on Signal and Image Processing (SIP)*, 2007.

[31] DeCoste, D., "Visualizing mercer kernel feature spaces via kernelized locally-linear embeddings," in *8th Intl. Conf. on Neural Information Processing*, 2001.

[32] Dhome, M., Richetin, M., and Lapreste, J.-T., "Determination of the attitude of 3d objects from a single perspective view," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 12, pp. 1265–1278, 1989.

[33] DoCarmo, M., *Diferential Geometry of Curves and Surfaces*. Prentice Hall, 1976.

[34] Doucet, A., deFreitas, N., and Gordon, N., *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[35] D.Ridder and Duin, R., "Locally linear embedding for classification," Tech. Rep. PH-2002-01, Pattern Recognition Group, Delft University of Technology, 2002.

[36] Drummond, T. and Cipolla, R., "Real-time tracking of multiple articulated structures in multiple views," in *Proc. 6th European Conf.on Computer Vision, ECCV*, pp. 20–36, 2000.

[37] Epstein, C. L. and Gage, M., "The curve shortening flow," *In Wave motion: theory, modelling, and computation (Math. Sci. Res. Inst. Publ.)*, vol. 7, pp. 15–59, 1987.

[38] Evans, L. C. and Spruck, J., "Motion of level sets by mean curvature.," *Int. Journal of Differential Geometry*, vol. 33, pp. 635–681, 1991.

[39] Evans, L. C. and Spruck, J., "Motion of level sets by mean curvature. ii.," *Transactions of Amer. Math. Society*, vol. 330, pp. 321–332, 1992.

[40] Evans, L. C. and Spruck, J., "Motion of level sets by mean curvature. iii.," *Journal Geom. Anal.*, vol. 2, pp. 121–150, 1992.

[41] Evans, L. C. and Spruck, J., "Motion of level sets by mean curvature. iv.," *Journal Geom. Anal.*, vol. 5, pp. 77–114, 1995.

[42] Faugeras, O., *Three-Dimensional Computer Vision, A Geometric Viewpoint.* MIT Press, 1993.

[43] Faugeras, O. and Gomes, J., "Dynamic shapes of arbitrary dimension: the vector distance functions," in *Proceedings of the Ninth IMA Conference on Mathematics of Surfaces* (Cipolla, R. and Martin, R., eds.), The Mathematics of Surfaces IX, Springer, 2000.

[44] Faugeras, O. D. and Keriven., R., "Variational principles, surface evolution pdes, level set methods and the stereo problem.," *INRIA Tech. report*, vol. 3021, pp. 1–37., 1996.

[45] Forsyth, D. and Ponce, J., *Computer Vision.* Prentice Hall, 2003.

[46] Funkhouser, T., Kazhdan, M., Shilane, P., Min, P., Kiefer, W., Tal, A., Rusinkiewicz, S., and Dobkin, D., "Modeling by example," in *ACM Transactions on Graphics (SIGGRAPH 2004)*, 2004.

[47] Gâteaux, R., "Sur les fonctionnelles continues et les fonctionnelles analytiques.," *Comptes rendus de l'academie des sciences*, vol. 157, 1913.

[48] Gelfand, I. M. and Fomin, S. V., *Calculus of Variations*. Dover.

[49] Gomes, J. and Faugeras, O., "Level sets and distance functions," vol. 1842 of *Lecture Notes in Computer Science*, pp. 588–602.

[50] Gomes, J. and Faugeras, O., "Reconciling distance functions and level sets," Tech. Rep. 3666, INRIA, 1999.

[51] Gomes, J. and Faugeras, O., "Reconciling distance functions and level sets," *Journal of Visual Communication and Image Representation*, vol. 11, no. 2, pp. 209–223, 2000.

[52] Gomes, J. and Faugeras, O., "Shape representation as the intersection of $n-k$ hypersurfaces," Tech. Rep. 4011, INRIA, 2000.

[53] Gonzalez, R. C. and Woods, R. E., *Digital Image Processing*. Addison-Wesley Longman., 2001.

[54] Grayson, M., "The heat equation shrinks embedded plane curves to round points," *Journal of Differential Geometry*, vol. 26, pp. 285–314, 1987.

[55] Grayson, M., "Shortening embedded curves," *Annals of Mathematics*, vol. 129, pp. 71–111, 1989.

[56] Hartley, R. and Zisserman, A., *Multiple view geometry in computer vision*. Cambridge University Press, 2000.

[57] Isard, M. and Blake, A., "Condensation – conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[58] Jackson, J., Yezzi, A., and Soatto, S., "Tracking deformable moving objects under severe occlusions," in *Conf. decision and control*, Dec, 2004.

[59] Jawahar, C., Biswas, P., and Ray, A., "Analysis of fuzzy thresholding schemes," *Pattern Recognition*, pp. 1339–1349, 2000.

[60] Julier, S. and Uhlmann, J., "A new method for the nonlinear transformation of means and covariances in filters estimators," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, 2000.

[61] Julier, S. and Uhlmann, J., "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–420, 2004.

[62] Kalman, R. E., "A new approach to linear filtering and prediction problems," *Transactions of the ASME - Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[63] Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., and Yezzi, A., "Conformal curvature flows: From phase transitions to active vision," *Archive for Rational Mechanics and Analysis*, vol. 134, no. 3, pp. 275–301, 1996.

[64] Kim, J., Fisher, J., Yezzi, A., Cetin, M., and Willsky, A., "Nonparametric methods for image segmentation using information theory and curve evolution," in *Proc. ICIP*, vol. 3, pp. 797–800, 2002.

[65] Kimmel, R., *Numerical Geometry of Images: Theory, Algorithm, and Applications*. Springer-Verlag, 2003.

[66] Kipnis, C. and Landim, C., *Scaling limits of interacting particle systems*. Springer-Verlag, New York, 1999.

[67] Kwok, J. and Tsang, I., "The pre-image problem in kernel methods," in *IEEE transactions on neural networks*, vol. 15, pp. 1517–1525.

[68] Kwok, J. and Tsang, I., "The pre-image problem in kernel methods," in *20th Intl. Conference on Machine Learning*, 2003.

[69] Lepetit, V. and Fua, P., "Monocular model-based 3d tracking of rigid objects: A survey," *Fundations and Trends in Computer Graphics and Vision*, vol. 1, no. 1, pp. 1–89., 2005.

[70] Leventon, M., Grimson, E., and Faugeras, O., "Statistical shape influence in geodesic active contours," in *Proc. CVPR*, pp. 1316–1324, IEEE, 2000.

[71] Li, P., Zhang, T., and Ma, B., "Unscented kalman filter for visual curve tracking," *Image and Vision Computing*, vol. 22, no. 2, pp. 157–164, 2004.

[72] M. Kass, A. W. and Terzopoulos, D., "Snakes: active contour models," *Int. Journal of Computer Vision*, vol. 1, pp. 321–331, 1987.

[73] Ma, Y., Soatto, S., Kosecka, J., and Sastry, S., *An invitation to 3D vision*. Springer.

[74] Malladi, R., Sethian, J. A., and Vemuri, B. C., "Shape modeling with front propagation: A level set approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, pp. 158–175, 1995.

[75] Marchand, E., Bouthemy, P., and Chaumette, F., "A 2d-3d model-based approach to real-time visual tracking," *Image and Vision Computing*, vol. 19, no. 13, pp. 941–955, 2001.

[76] Melonakos, J., Pichon, E., Angenent, S., and Tannenbaum, A., "Finsler active contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.

[77] Mercer, J., "Functions of positive and negative type and their connection with the theory of integral equations," in *Philos. Trans. Roy. Soc. London*, vol. 209, pp. 415–446, 1909.

[78] Michailovich, O., Rathi, Y., and Tannenbaum, A., "Image segmentation using active contours driven by the bhattacharyya gradient flow," *IEEE Transactions on Image Processing*, pp. 2787–2801, 2007.

[79] Michor, P. and Mumford, D., "Riemannian geometries of space of plane curves," *http://front.math.ucdavis.edu/math.DG/0312384*.

[80] Mika, S., Schölkopf, B., Smola, A. J., Müller, K.-R., Scholz, M., and Rätsch, G., "Kernel PCA and de–noising in feature spaces," in *Advances in Neural Information Processing Systems*, vol. 11, MIT Press, 1999.

[81] Morel, J.-M. and Solimini, S., *Variational Methods for Image Segmentation*. Birkhauser, 1994.

[82] Mourragui, M., "Comportement hydrodynamique et entropie relative des processus de sauts, de naissances et de morts," *Ann. Inst. H. poincaré, Prob. & Stat*, vol. 32, pp. 361–385, 1996.

[83] Mumford, D. and Shah, J., "Optimal approximation by piecewise smooth functions and associated variational problems," *Commun. Pure Applied Mathematics*, vol. 42, pp. 577–685, 1989.

[84] Nain, D., Haker, S., Bobick, A. F., and Tannenbaum, A., "Multiscale 3-d shape representation and segmentation using spherical wavelets," *IEEE Trans. Med. Imaging*, vol. 26, no. 4, pp. 598–618, 2007.

[85] Nain, D., S.Haker, A.Bobick, and A.Tannenbaum., "Shape-driven 3d segmentation using spherical wavelets," in *Proc. MICCAI*, vol. 1, pp. 66–74, 2006.

[86] Niethammer, M. and Tannenbaum, A., "Dynamic geodesic snakes for visual tracking," in *Proc. CVPR*, vol. 1, pp. 660–667, 2004.

[87] Osher, S. and Fedkiw, R., *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag, 2003.

[88] Osher, S. and Paragios, N., *Geometric Level Set Methods in Imaging, Vision and Graphics*. Springer Verlag, 2003.

[89] Osher, S. J. and Sethian, J. A., "Fronts propagation with curvature dependent speed: Algorithms based on hamilton-jacobi formulations," *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.

[90] Paragios, N., Chen, Y., and Faugeras, O., *Handbook of Mathematical Models in Computer Vision*. Springer, 2005.

[91] Paragios, N. and Deriche, R., "Geodesic active regions for supervised texture segmentation," in *ICCV (2)*, pp. 926–932, 1999.

[92] Paragios, N. and Deriche, R., "Geodesic active contours and level sets for the detection and tracking of moving objects," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 3, pp. 266–280, 2000.

[93] Paragios, N. and Deriche, R., "Geodesic active regions: A new paradigm to deal with frame partition problems in computer vision," *Journal of Visual Communication and Image Representation*, vol. 13, pp. 249–268, 2002.

[94] Parzen, E., "On the estimation of a probability density function and the mode," *Annals of Mathematical Statistics*, vol. 33, pp. 1065–1076, 1962.

[95] Peterfreund, N., "Robust tracking of position and velocity with Kalman snakes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 564–569, 1999.

[96] Peterfreund, N., "The velocity snake: deformable contour for tracking in spatio-velocity space," *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 346–356, 1999.

[97] Quan, L. and Lan, Z.-D., "Linear n-point camera pose determination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 774–780, 1999.

[98] Rathi, Y., Dambreville, S., and Tannenbaum., A., "Comparative analysis of kernel methods for statistical shape learning," in *CVAMIA*, pp. 96–107, 2006.

[99] Rathi, Y., Dambreville, S., and Tannenbaum., A., "Statistical shape analysis using kernel pca," in *Proceedings of SPIE Electronic Imaging*, vol. 6064, pp. 425–432, 2006.

[100] Rathi, Y., Michailovich, O., and Tannenbaum., A., "Seeing the unseen: Segmenting with distributions," in *Intl. Conf. Signal and Image Processing*, vol. 534, 2006.

[101] Rathi, Y., Vaswani, N., Tannenbaum, A., and Yezzi, A., "Particle filtering for geometric active contours with application to tracking moving and deforming objects," in *CVPR 2005*.

[102] Rathi, Y., Dambreville, S., Niethammer, M., Tannenbaum, A., Shenton, M., and Levitt, J., "Segmenting images analytically in shape space," in *SPIE Medical Imaging*, 2008.

[103] Rosenhahn, B., Brox, T., and Weickert, J., "Three-dimensional shape knowledge for joint image segmentation and pose tracking," *International Journal of Computer Vision (IJCV)*, vol. 73, no. 3, pp. 243–262., 2007.

[104] Rosenhahn, B., Perwass, C., and Sommer, G., "Pose estimation of free-form contours," *Int. Journal of Computer Vision (IJCV)*, vol. 62, no. 3, pp. 267–289., 2005.

[105] Rousson, M. and Cremers, D., "Efficient Kernel Density Estimation of Shape and Intensity Priors for Level Set Segmentation," in *Medical Image Computing and Computer Assisted Intervention*, vol. 1, pp. 757–764, 2005.

[106] Rousson, M. and Deriche, R., "A variational framework for active and adaptative segmentation of vector valued images," in *Proc. of the Workshop on Motion and Video Computing*, p. 56, 2002.

[107] Rousson, M. and Paragios, N., "Shape priors for level set representations," in *Proceedings of European Conference on Computer Vision*, pp. 78–92, 2002.

[108] Rousson, M. and Paragios, N., "Prior knowledge, level set representations and visual grouping," *International Journal of Computer Vision*, 2008.

[109] Sapiro, G., ed., *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2001.

[110] Sapiro, G., ed., *Geometric Partial Differential Equations and Image Analysis*. Cambridge Press, 20010.

[111] Saul, L. K. and Roweis, S., "An introduction to locally linear embedding," *http://www.cs.toronto.edu/ roweis/lle/papers/lleintro.pdf*.

[112] Scholkopf, B., "The kernel trick for distances," NIPS, pp. 301–307, 2000.

[113] Schölkopf, B., Mika, S., and Müller, K., "Nonlinear component analysis as a kernel eigenvalue problem," in *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

[114] Schölkopf, B., Mika, S., and Müller, K., "An introduction to kernel-based learning algorithms," in *IEEE Trans. on Neural Networks*, vol. 12, pp. 181–201, 2001.

[115] Sethian, J. A., "A review of recent numerical algorithms for hypersurfaces moving with curvature dependent speed," *J. Differential Geometry*, vol. 31, pp. 131–161, 1989.

[116] Sethian, J. A., *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 2nd ed., 1999.

[117] Sezgin, M. and Sankur, B., "Survey over image thresholding techniques ad quantitaive performance evaluation," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 146–165, 2004.

[118] Shapiro, L. G. and Stockman, G. C., *Computer Vision*. Prentice-Hall, 2001.

[119] Slabaugh, G., Unal, G., Fang, T., and Wels, M., " Ultrasound-Specific Segmentation via Decorrelation and Statistical Region-Based Active Contours

," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06)* , pp. 45–53, 2006.

[120] Srivastava, A., Joshi, S., Mio, W., and Liu, X., "Statistical shape analysis: Clustering, learning and testing," *Trans. PAMI*, 2005.

[121] Taron, M., Paragios, N., and Jolly, M.-P., "Registration with uncertainties and statistical modeling of shapes with variable metric kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.

[122] Terzopoulos, D. and Szeliski, R., *Active Vision*, ch. Tracking with Kalman Snakes, pp. 3–20. MIT Press, 1992.

[123] Tsai, A. and Yezzi, A., "Model-based curve evolution technique for image segmentation," in *Proceedings of Computer Vision and Pattern Recognition*, vol. 1.

[124] Tsai, A., Yezzi, T., Wells, W., Tempany, C., Tucker, D., Fan, A., Grimson, E., and Willsky, A., "A shape-based approach to the segmentation of medical imagery using level sets," *IEEE Trans. on Medical Imaging*, vol. 22, no. 2, pp. 137–153, 2003.

[125] Unal, G., Yezzi, A., Soatto, S., and Slabaugh, G., "A variational approach to problems in calibration of multiple cameras.," *Trans. Pattern Analysis and Machine Intelligence (T. PAMI)*, vol. 29, no. 8, pp. 1322–1338., 2007.

[126] Unal, G. B., Nain, D., Ben-Arous, G., Shimkin, N., Tannenbaum, A., and Zeitouni, O., "Algorithms for stochastic approximations of curvature flows," in *Proceedings of ICIP*, vol. 2, pp. 651–654, 2003.

[127] van der Merwe, R. and Wan, E., "The unscented kalman filter for nonlinear estimation," *Proceedings of IEEE Symposium*, 2000.

[128] van der Merwe, R. and Wan, E., "The square-root unscented kalman filter for state and parameter estimation," *Proceedings of the International Conference on Accoustics, speech, and Signal Processing(ICASSP)*, 2001.

[129] Wang, Y. and Staib, L., "Boundary finding with correspondance using statistical shape models," in *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 338–345, 1998.

[130] Yezzi, A., Kichenassamy, S., and Kumar, A., "A geometric snake model for segmentation of medical imagery," in *IEEE Trans. Medical Imag.*, vol. 16, pp. 199–209, 1997.

[131] Yezzi, A. and Mennucci, A., "Metrics in the space of curves," *http://arxiv.org/abs/math.DG/0412454*, 2004.

[132] Yezzi, A. and Soatto, S., "Deformotion: Deforming motion, shape average and the joint registration and approximation of structures in images," *International Journal of Computer Vision*, vol. 53, no. 2, pp. 153–167, 2003.

[133] Yezzi, A. and Soatto, S., "Stereoscopic segmentation.," *International Journal of Computer Vision (IJCV)*, vol. 53, no. 3, pp. 31–43., 2003.

[134] Yezzi, A. and Soatto, S., "Structure from motion for scenes without features.," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 171–178, 2003.

[135] Yezzi, A., Tsai, A., and Willsky, A., "Medical image segmentation via coupled curve evolution equations with global constraints," in *Proc. Workshop on Mathematical Methods in Biomedical Image Analysis*, pp. 12–19, 2000.

[136] Yuille, A., Cohen, D., and Halliman, P., "Feature extraction from faces using deformable templates," in *Proc. CVPR*, pp. 104–109, IEEE, 1989.

[137] Zerroug, M. and Nevatia, R., "Pose estimation of multi-part curved objects," in *ISCV '95: Proceedings of the International Symposium on Computer Vision*, p. 431, 1995.

[138] Zhang, T. and Freedman, D., "Tracking objects using density matching and shape priors," in *Proceedings of the Ninth International Conference on Computer Vision*, pp. 1950–1954, IEEE, 2003.

[139] Zhu, S. C. and Yuille, A. L., "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884–900, 1996.

VITA

Samuel Dambreville received the "Diplome d'ingenieur en Genie Electrique et Automatique" (M.Eng.) from the ENSEEIHT Toulouse, France, in 2003. He also holds a Master of Science in Electrical Engineering as well as a Master in Business Administration from the Georgia Institute of Technology, which he received in 2003 and 2006, respectively. His PhD research interests include the robust tracking of deformable objects, and the development of statistical and shape-driven techniques for active contour segmentation. He was the recipient of the Hugo O. Schuck Best Paper Award at the 2007 American Control Conference (along with Yogesh Rathi and Allen Tannenbaum).